

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Revue critique des normes ISO en exigences de qualité du point de vue de l'utilisabilité

D'Agostino, Christophe

Award date:
2016

Awarding institution:
Université de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Facultés Universitaires de Namur
Faculté d'informatique
Année académique 2015-2016

Revue critique des normes ISO en exigences de qualité du point de vue de l'utilisabilité

Christophe D'AGOSTINO

Mémoire présenté en vue de l'obtention du grade de master
en sciences informatiques

Résumé

Il existe de nombreuses normes relatives à la notion d'utilisabilité, dont, notamment, ISO 9241 qui est consacrée à l'ergonomie de l'interaction homme-système.

L'Organisation Internationale de Normalisation, ou l'ISO, est occupé de publier ISO/IEC 25000, aussi connue sous le nom de SQuaRE (Ingénierie des systèmes et du logiciel – Exigences de qualité et évaluation des systèmes et du logiciel), qui définit des caractéristiques et sous-caractéristiques de qualité non-fonctionnelle ainsi que des mesures.

Le but du présent mémoire est, dans une première phase, de faire le point sur ce qui existe actuellement en analysant les travaux qui ont déjà été menés sur le sujet.

Dans une seconde phase, nous allons tracer des liens entre ISO 9241 et les modèles de qualité proposées par ISO/IEC 25000.

Enfin, pour les caractéristiques et sous-caractéristiques traitant de l'utilisabilité, nous faisons une critique sur les mesures proposées. Nous allons voir si elles sont adaptées, faciles à mettre en place et, le cas échéant, proposerons une mesure adaptée.

Du fait qu'ISO 25000 est une norme assez récente, il n'existe pas encore, à l'heure actuelle, de publication scientifique proposant ce genre d'approche. La principale valeur ajoutée de ce mémoire est donc de faire une revue critique entre les normes relatives à l'utilisabilité et ISO 25000.

Mots clés

Utilisabilité, Ergonomie, Normalisation, ISO 9241, ISO 25000, SQuaRE, Qualité

Abstract

There are a lot of standards related to the notion of usability, including, among them, ISO 9241, which is dedicated to the ergonomics of human-computer interaction.

The International Organization for Standardization, or ISO, is as well dealing with the publication of ISO/IEC 25000, also known as SQuaRE (Systems and Software Engineering – Systems and software Quality Requirements and Evaluation), that defines non-functional quality characteristics and sublevel characteristics as well as metrics.

The goal of this master thesis is, first, to point out what is currently effective by analysing actual works on this subject.

Then, we will try to match links between ISO 9241 and ISO/IEC 25000 quality standards models.

Eventually, concerning usability characteristics and sublevel characteristics, we will discuss about offered metrics. We will see if they are adapted, easy to implement and if necessary, we will propose an appropriate measure.

The main added value of this master thesis is that it compares standards relative to usability and ISO/IEC 25000 that is quite new, and so far, there is no such scientific publication that offers this approach.

Keywords

Usability, Ergonomics, Normalisation, ISO 9241, ISO 25000, SQuaRE, Quality

Avant-propos

Ce mémoire est le résultat de plusieurs mois de travail et rentre dans le cadre de l'obtention du Master 60 en Sciences Informatiques à horaire décalé.

Ayant déjà une expérience dans le domaine de la qualité informatique, qui, je dois le dire, me passionne, j'ai souhaité orienter mon travail vers ce domaine de recherche.

Je souhaitais proposer un mémoire portant sur SQuaRE, norme de l'Organisation Internationale de Normalisation, qui définit des critères de qualité non-fonctionnelle (c'est la norme qui succède à ISO 9126). Le champ de recherche autour de ce domaine est bien évidemment gigantesque. C'est suite aux discussions que j'ai pu avoir avec Messieurs *Bruno Dumas* et *Benoît Vanderose*, que j'ai décidé de me concentrer sur la notion d'utilisabilité. Je les remercie d'ailleurs pour leur encadrement et leurs précieux conseils qu'ils m'ont apporté tout au long de cette année.

Le travail présenté dans ce mémoire n'aurait jamais pu se faire sans le support et le soutien d'autres intervenants. Je tiens donc à remercier ceux qui y ont contribué, de près ou de loin.

Merci à la société *Aubay.Promotic*, de m'avoir permis, au travers de mes diverses expériences, de trouver ma voie dans le domaine de la qualité et du test logiciel, ce qui a indirectement aiguillé le sujet du présent mémoire.

Merci à *David Clément*, collègue et ami, sans qui tout cela n'aurait pas été possible. Nous avons décidé, en 2014, de nous lancer ensemble dans cette aventure de reprise des cours à horaire décalé.

Merci à *Fabian Dermine*, chef du service informatique au SPW-DGO3, pour son soutien, sa confiance et son suivi.

Merci à mes collègues, qui m'ont encouragé tout au long de mon parcours. Tout particulièrement à *Gilles Ronvaux* et *Philippe Vervier* pour leur relecture.

Merci à *Ludovic Coletti* pour le « petit coup de pouce » dans mes tâches de traduction.

Merci à *Benjamine Lurquin* pour ses conseils pratiques et sa gentillesse au jour le jour.

Merci à *mes parents* qui m'ont encouragé tout au long de mes études et d'avoir fait de moi une grande partie de ce que je suis devenu.

Enfin, sans doute l'hommage qui me tient le plus à cœur, je tiens à remercier ma compagne *Virginie Wislet*, grâce à son soutien et aux sacrifices concédés ces deux dernières années, elle m'a permis de vivre l'aventure, à la fois passionnante et éprouvante, que sont les études en horaire décalé.

Table des matières

Résumé.....	1
Abstract	2
Avant-propos.....	3
Table des matières	4
Table des illustrations	7
Abréviations	8
Introduction.....	9
Chapitre 1 : Les normes.....	11
1.1 Un peu d'histoire.....	11
1.2 Qu'est-ce qu'une norme ?.....	13
1.3 Les avantages des normes internationales	14
1.4 Les inconvénients des normes internationales	15
1.5 Comment l'ISO élabore-t-elle les normes ?	16
Chapitre 2 : L'utilisabilité.....	17
2.1 Utilité, utilisabilité et ergonomie	17
2.2 Enjeux de l'utilisabilité	18
2.3 Heuristiques de Nielsen.....	19
2.4 Conception centrée sur l'utilisateur.....	20
2.5 Standards en relation avec l'utilisabilité	21
2.5.1 Classification.....	22
2.5.2 Evolution des normes.....	24
Chapitre 3 : ISO 9241 – Ergonomie de l'interaction homme-système.....	26
3.1 Organisation de la norme.....	26
3.2 Lien entre les parties 11, 110 et 12	28
3.3 Focus sur les parties d'ISO 9241.....	29
Chapitre 4 : ISO 25000 - Exigences de qualité des systèmes et du logiciel et évaluation	30
4.1 ISO 9126 - Technologies de l'Information : Qualités des produits logiciels	30
4.2 ISO 14598 - Technologies de l'information : Évaluation du produit logiciel	33
4.3 SQuaRE	34
4.4 Organisation de la norme.....	35
4.4.1 ISO/IEC 2500n – Quality Manangement Division.....	35
4.4.2 ISO/IEC 2501n – Quality Model Division	35
4.4.3 ISO/IEC 2502n – Quality Measurement Division.....	35
4.4.4 ISO/IEC 2503n- Quality Requirement Division	35
4.4.5 ISO/IEC 2504n – Qualityt Evaluation Division	36

4.4.6 ISO/IEC 25050 to 25099 – SQuaRE extensions	36
4.5 Modèles de qualité.....	37
4.5.1 Qualité à l'Usage.....	37
4.5.2 Qualité Produit	38
4.6 Caractéristiques concernées par l'utilisabilité	38
Chapitre 5 : Croisement des normes.....	39
5.1 Lien entre ISO 9241-11:1998 et ISO 25010:2011	40
5.1.1 Efficacité	40
5.1.2 Efficience	41
5.1.3 Satisfaction	41
5.2 Lien entre ISO 9241-110:2006 et ISO 25010:2011	42
5.2.1 Adaptation à la tâche	43
5.2.2 Aptitude à l'individualisation.....	43
5.2.3 Facilité d'apprentissage.....	44
5.2.4 Caractère autodescriptif.....	45
5.2.5 Contrôle utilisateur	45
5.2.6 Conformité aux attentes de l'utilisateur	46
5.2.7 Tolérance à l'erreur	46
5.3 Mapping entre ISO 9241 et ISO 25000.....	47
5.4 Les caractéristiques d'ISO 25010.....	49
5.5 Cas pratique.....	49
Chapitre 6 : Mesurer l'utilisabilité.....	50
6.1 Métriques concernant l'utilisabilité	52
6.1.1 Les métriques de compréhensibilité	54
6.1.2 Les métriques de facilité d'apprentissage.....	55
6.1.3 Les métriques d'opérabilité.....	56
6.1.4 Les métriques d'attraction	58
6.1.5 Les métriques de respect de l'utilisabilité.....	59
6.2 Critique des métriques d'ISO 9126.....	60
6.3 Tester l'utilisabilité	61
6.3.6 Questionnaires	61
6.3.7 Think aloud protocol	61
6.3.8 Prototypage.....	61
6.3.9 Eye Tracking.....	62
Conclusion	63
Références.....	65

Bibliographie	65
Webographie.....	67
Annexe 1 : SQuaRE - Quality In Use.....	68
Annexe 2 : SQuaRE - System/Software Product Quality	69
Annexe 3 : Rapport sur la qualité d'un produit.....	70

Table des illustrations

Figure 1.1 : La répartition des normes ISO fin 2011 montre une grande variété de secteurs et sujets techniques.....	12
Figure 1.2 : Processus d'élaboration d'une norme	16
Figure 2.1 : Current Research - Usability Standards – XBOSOFT	21
Figure 2.2 : Classement des standards relatifs à l'utilisabilité en catégories.....	22
Figure 3.1 : Organisation de la norme ISO 9241.....	27
Figure 3.2 : Lien entre 9241-11, 9241-110 et 9241-12.....	28
Figure 4.1 : SQuaRE = ISO 9126 + ISO 14598.....	30
Figure 4.2 : Organisation de la norme ISO 9126.....	30
Figure 4.3 : ISO 9126 - Caractéristiques internes et externes d'un produit.....	31
Figure 4.4 : Organisation de la norme ISO 14598.....	33
Figure 4.5 : L'évaluation du produit logiciel s'appuie sur les normes.....	33
Figure 4.6 : Organisation de la norme ISO 25000.....	35
Figure 4.7 : ISO 25000 - Modèle Qualité à l'usage	37
Figure 4.8 : ISO 25000 - Modèle Qualité Produit	38
Figure 5.1 : Caractéristique d'ISO 9241-11 et le modèle à l'usage d'ISO 25010	40
Figure 5.2 : Caractéristique d'ISO 9241-110 et ISO 25010	42
Figure 5.3 : Mapping entre ISO 9241 et ISO 25000	48
Figure 6.1 : User Needs Characteristics & Weights.....	50
Figure 6.2 : Structure of the quality Measurement Division.....	51
Figure 6.3 : Example systems within the system complexity space (interaction complexity vs. domain complexity).....	52
Figure 6.4 : ISO 9126-1 Relationship between types of metrics	53

Abréviations

Abréviation	Signification
CMMI	C apability M aturity M odel I ntegration
HCI	H uman- C omputer I nteraction (interaction homme-machine)
IEC	C ommission E lectrotechnique I nternationale (CEI, ou IEC en anglais)
IEEE	I nstitute of E lectrical and E lectronics E ngineers
ISO	I nternational O rganization for S tandardization (Organisation internationale de normalisation)
NTIC	N ouvelles T echnologies de l' I nformation et de la C ommunication
SEI	S oftware E ngineering I nstitute
SQuaRE	S ystems and software Q uality R equirements and E valuation (Exigences de qualité des systèmes et du logiciel et évaluation)
TR	T echnical R eport (Rapport technique)
UCD	U ser- C entered D esign (conception centrée sur l'utilisateur ou conception orientée utilisateur)

Introduction

La qualité a toujours suscité l'intérêt aussi bien des producteurs de produits industriels que des développeurs de produits logiciels afin d'assurer la satisfaction de leurs clients et affronter la concurrence internationale. En génie logiciel, la qualité du logiciel a été reconnue comme un domaine de connaissance distinct des différents domaines de connaissance du « Guide au corpus des connaissances en génie logiciel » (Guide to the Software Engineering Body of Knowledge - SWEBOK-ISO TR 19759).

Les standards les plus populaires traitant de la qualité des produits sont ISO 9126 (Génie logiciel : qualité des produits) et ISO 25000 (Ingénierie des systèmes et du logiciel : Exigences de qualité des systèmes et du logiciel et évaluation). Ils fournissent un modèle, fait de caractéristiques et sous-caractéristiques, permettant de lier les qualités externes¹ d'un logiciel à ses qualités internes², et introduisent le concept de « Qualité à l'usage ».

Un autre paradigme, également très important, et que l'on retrouve aujourd'hui au cœur des processus de développement logiciel est celui de la conception centrée utilisateur.

Toutes les techniques, méthodes, et processus de développement utilisés visent à connaître et comprendre les utilisateurs (analyser leurs besoins, évaluer leurs manières d'utiliser les systèmes) dans le but de concevoir et développer des systèmes utilisables, c'est-à-dire, en adéquation avec leurs comportements, leurs compétences et leurs besoins. Ces techniques, méthodes, et processus se perfectionnent au fur et à mesure et se répandent progressivement dans les industries. La philosophie sous-jacente à la conception centrée utilisateur est de concevoir des systèmes pouvant être utilisés efficacement, avec satisfaction par les utilisateurs, et ce dès leur premier contact avec le système (et donc généralement sans avoir suivi de formation à l'utilisation du système). La mise en œuvre d'une telle conception a pour but de produire des systèmes interactifs satisfaisant la propriété d'utilisabilité. Cette propriété a été normalisée (ISO 9241 – Ergonomie de l'interaction homme-système) et se décompose en 3 facteurs principaux « efficacité », « efficience » et « satisfaction ». Ces trois facteurs sont abordés de façon intégrée dans la conception centrée utilisateur en plaçant les utilisateurs potentiels au cœur même du processus de développement.

Les normes ISO sont nombreuses et en constante évolution. Elles sont revues, de nouvelles normes apparaissent, remplacent des normes existantes ou encore certaines parties de normes viennent les compléter. Mais elles n'évoluent pas toutes au même rythme.

Le but de ce mémoire est de se pencher sur les différents standards proposés par l'ISO concernant ces deux paradigmes : la qualité logicielle et l'utilisabilité.

Sachant que le standard ISO 9241-11 (Lignes directrices concernant l'utilisabilité) est sorti en 1998, nous aurons l'occasion de voir si la définition donnée à l'utilisabilité n'a pas évolué depuis.

Le standard ISO 25010 (Modèles de qualité du système et du logiciel), qui est une sous-partie d'ISO 25000, est plus récent car il a été publié en 2011. Nous verrons donc dans quelle mesure celui-ci a pris en considération ce qui est défini dans ISO 9241.

¹ La qualité externe d'un produit logiciel est la qualité vue de l'extérieur (en cours d'utilisation), du point de vue de l'utilisateur. Le logiciel doit être capable de répondre aux besoins exprimés et implicites d'utilisation du système.

² La qualité interne d'un produit logiciel est la qualité vue de l'intérieur (statique), du point de vue du concepteur. Le logiciel satisfait les spécifications prédéfinies, et répond aux besoins exprimés et implicite du logiciel.

Le sujet de ce mémoire vise trois objectifs :

- Le premier objectif va être de faire le point sur les documents normatifs concernant l'utilisabilité et la qualité logiciel proposés par l'Organisation Internationale de Normalisation. Ces documents, en plus d'être nombreux, ne sont pas tous à jour et évoluent différemment les uns par rapport aux autres. Nous allons donc tenter d'y voir un peu plus clair en brossant un historique de ces normes et en essayant de comprendre leur utilité.
- Le deuxième objectif vise à mettre ces normes en rapport et voir s'il existe des liens entre elles. Pour cela, nous allons nous pencher sur la norme la plus récente concernant la qualité logiciel, SQuaRE, et vérifier dans quelle mesure elle tient compte d'ISO 9241 lorsqu'elle dresse des caractéristiques en rapport avec l'utilisabilité d'un logiciel. Du fait que SQuaRE est une norme assez récente, ce genre d'approche n'a pas encore été proposé.
- Le troisième objectif est d'identifier les mesures proposées par l'ISO concernant l'utilisabilité et de vérifier leurs pertinences.

La suite du présent mémoire est structurée comme suit :

- Chapitre 1 : Les normes
- Chapitre 2 : L'utilisabilité
- Chapitre 3 : ISO 9241 – Ergonomie de l'interaction homme-système
- Chapitre 4 : ISO 25000 - Exigences de qualité des systèmes et du logiciel et évaluation
- Chapitre 5 : Croisement des normes
- Chapitre 6 : Mesurer l'utilisabilité

Chapitre 1 : Les normes

Avant toute chose, nous allons, dans ce chapitre, introduire la notion de norme. Nous allons voir à quoi elles servent et nous allons nous pencher plus particulièrement sur l'ISO (puisque les normes abordées dans ce mémoire sont celles d'ISO).

Nous verrons également comment ces normes sont mises en place, leur processus d'élaboration et qui décide de la création ou de la modification d'une norme.

1.1 Un peu d'histoire

Tout au long de l'histoire, nous pouvons retrouver des esquisses de normalisation qui ont accompagné de grands mouvements de diffusion et de conquête.

Nous pouvons, à titre d'exemple, citer l'empire romain qui a tenté d'imposer nombre de ses standards aux pays conquis (standards de construction et d'administration dont, par exemple, les voies romaines).

Citons également, au Moyen-Âge, le système des corporations qui imposait à certains corps de métier des réglementations strictes comprenant notamment les normes de qualité des produits ou des services.

C'est au 19^{ème} siècle, dans le domaine industriel, qu'un très grand effort de normalisation a été entrepris par les principaux pays industrialisés.

Cette normalisation industrielle est née dans les milieux de l'industrie et du commerce, du contexte de guerre, puis de reconstruction qui a suivi la Première Guerre mondiale.

Concernant l'ISO, nous pouvons citer les dates clés suivantes :

- 1946, Londres : 65 délégués de 25 pays se réunissent pour envisager l'avenir de la normalisation internationale.
- 1951 : Apparition de la première norme ISO (ISO/R 1:1951, Température normale de référence des mesures industrielles de longueur).
- 1957 : l'ISO voit officiellement le jour et crée 67 comités techniques (groupes d'experts travaillant sur un sujet spécifique). L'ISO est le premier producteur mondial de normes internationales d'application volontaire.
- 1970 : le Secrétaire général de l'ISO, Olle Sturen, s'attache à faire de l'ISO une organisation véritablement internationale.
- 1987 : l'ISO publie sa première norme relative au management de la qualité. Les normes de la famille ISO 9000 font partie des normes ISO les plus connues et les plus vendues.
- 2005 : le comité technique mixte de l'ISO et de IEC lance la norme ISO/IEC 27001 relative aux systèmes de management pour la sécurité de l'information. A l'heure où les entreprises s'appuient de plus en plus sur les technologies de l'information, la sécurisation des systèmes et la réduction des risques gagnent en importance.

Aujourd'hui, comme le montre la Figure 1.1, les normes internationales de l'ISO couvrent la quasi-totalité des aspects techniques et économiques.

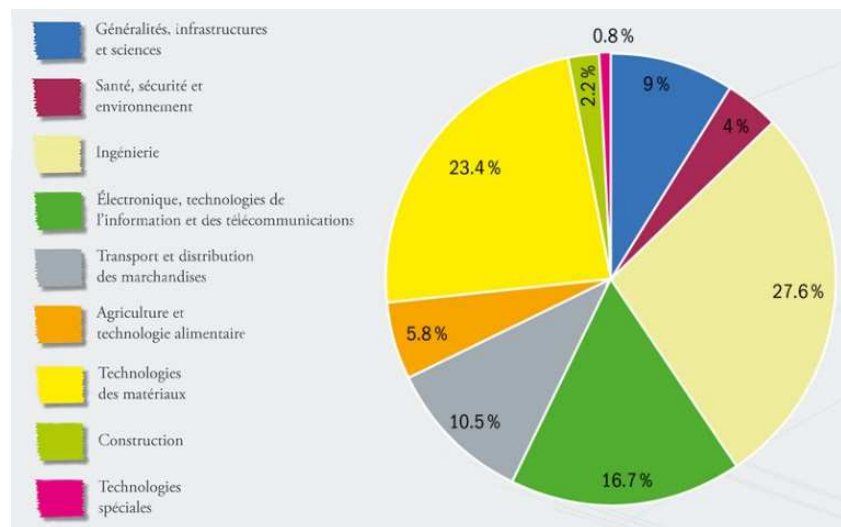


Figure 1.1 : La répartition des normes ISO fin 2011 montre une grande variété de secteurs et sujets techniques

1.2 Qu'est-ce qu'une norme ?

Le terme « norme » désigne un document établi à un certain niveau : national, régional ou international.

La norme est un document de référence, volontaire, élaboré de façon consensuelle, collective et participative, dont le but est de faciliter les échanges.

Elle permet de :

- formaliser le savoir-faire et limiter l'explosion anarchique de la variété,
- disposer d'un langage commun, partagé et universel,
- harmoniser les relations entre un client et ses fournisseurs,
- bénéficier d'un cadre de référence permettant à l'entreprise de progresser.

Si nous reprenons la définition de l'Organisation Internationale de Normalisation, une norme est un document qui définit des exigences, des spécifications, des lignes directrices ou des caractéristiques à utiliser systématiquement pour assurer l'aptitude à l'emploi des matériaux, produits, processus et services.

Règle fixant les conditions de la réalisation d'une opération, de l'exécution d'un objet ou de l'élaboration d'un produit dont on veut unifier l'emploi ou assurer l'interchangeabilité.

Larousse

En d'autres termes, les normes sont des rouages indispensables qui établissent des spécifications de premier ordre pour les produits, les services et les systèmes dans une optique de qualité, de sécurité et d'efficacité.

Une norme associe, à l'état de l'art, la synthèse des bonnes pratiques des entreprises qui ont participé à la normalisation. Lorsqu'une norme est ajustée par l'entreprise à ses besoins propres, on parle alors de standard d'entreprise.

1.3 Les avantages des normes internationales

Le monde dans lequel nous vivons offre autant de possibilités qu'il y a d'idées. Mais le monde est parfois complexe et demande une rationalisation.

Grâce aux normes ISO, le monde devient plus sûr, plus sain et plus efficace sur bien des plans. De la sécurité alimentaire à l'informatique, de la santé aux nouvelles technologies, notre environnement, notre économie et notre société doivent faire face à de nombreux défis. Parfois sans en être conscient, l'ISO améliore notre quotidien en mobilisant les meilleurs expertises et compétences internationales.

Dans notre monde en constante évolution, les normes aident à stimuler la croissance, à ouvrir de nouveaux marchés et à rendre le commerce plus équitable. L'ISO nous aide à relever les défis mondiaux tel que le réchauffement climatique, la sécurité routière, l'énergie et la responsabilité sociétale.

Les normes ISO touchent pratiquement tout ce que nous faisons. Grâce à elles, nous restons connectés, nous nous distrayons, nous sommes plus productifs, plus créatifs et nous échangeons des idées tout en veillant à notre sécurité et notre santé.

L'ISO est le plus grand fondateur de normes internationales au monde. Avec à son actif plus de 20500 normes couvrant pratiquement chaque aspect de la technologie et des affaires depuis plus de 60 ans.

Les normes internationales offrent de nombreux avantages autant sur le plan technique, économique que sociétal. Elles permettent d'harmoniser les spécifications techniques des produits et des services en renforçant l'efficacité de l'industrie et en éliminant les obstacles au commerce international.

La conformité aux normes internationales est un gage de confiance pour les consommateurs que les produits sont sûrs, efficaces et sans danger pour l'environnement.

Pour les entreprises, les normes internationales sont des outils stratégiques et des lignes directrices destinés à aider les entreprises à relever les défis les plus urgents pour l'économie moderne. Elles garantissent l'efficacité optimale des opérations commerciales, augmentent la productivité et aident les entreprises à accéder à de nouveaux marchés.

Parmi les nombreux avantages, citons ceux-ci :

- **Economie de coûts** – les normes internationales aident à optimiser les opérations (impact positif sur le bilan) ;
- **Renforcement de la satisfaction clientèle** – les normes internationales aident à améliorer la qualité, à renforcer la satisfaction clientèle et à accroître les ventes ;
- **Accès à de nouveaux marchés** – les normes internationales aident à éliminer les obstacles au commerce et à ouvrir l'accès aux marchés mondiaux ;
- **Augmentation des parts de marché** – les normes internationales aident à accroître la productivité et les avantages concurrentiels ;
- **Avantages environnementaux** – les normes internationales aident à réduire les effets négatifs sur l'environnement.

1.4 Les inconvénients des normes internationales

Puisque nous avons parlé des avantages et vanté les bienfaits d'ISO, nous devons également citer quelques désavantages des normes ISO.

Comme nous avons eu l'occasion de le voir, il existe un très grand nombre de normes couvrant pratiquement chaque aspect de la technologie et donc, inévitablement, il devient très difficile de s'y retrouver. Le monde ISO manque cruellement d'une typologie permettant à quelqu'un de plus facilement cibler la partie recherchée. C'est d'ailleurs ce que propose ce mémoire en faisant le point sur les normes relatives à l'utilisabilité.

D'autant plus que le domaine de la technologie évolue très vite alors que les normes mettent des années pour être créées, revues, remplacées ou encore déclassées. Il se creuse donc toujours un écart entre les normes et les pratiques utilisées dans les entreprises.

Celui qui revient le plus souvent auprès de bon nombre de dirigeants de PME est le coût de la démarche. Le budget à consacrer à la normalisation est très variable. Il dépend essentiellement de la taille de la société, du nombre de sites qu'elle regroupe ou des produits à certifier.

Un autre désavantage des normes ISO est qu'elles sont souvent lourdes et peu concrètes. La plupart de celles-ci donnent des lignes directrices, mais restent très haut niveau.

Si l'on souhaite se conformer aux normes, deux approches sont possibles :

- La démarche « normative »¹ : Cette méthode conduit inévitablement à des inadaptations et à des lourdeurs.
- La démarche « fonctionnelle »² : Cette méthode permet, également, d'atteindre l'équivalent d'une certification de service, mais sans la rigidité et la lourdeur associées.

¹ Démarche permettant d'atteindre la conformité aux normes en reprenant celles-ci article par article et en faisant le nécessaire pour se mettre en conformité.

² Démarche axée sur l'amélioration de l'entreprise, en appliquant la norme de façon très adaptée et sans chercher la « conformité » officielle.

1.5 Comment l'ISO élabore-t-elle les normes ?

Une des principales forces des normes ISO est qu'elles sont créées par ceux-là même qui en ont besoin. Les experts de l'industrie animent toutes les phases de leur processus d'élaboration, de la décision sur la nécessité d'une nouvelle norme à la définition de l'ensemble du contenu technique.

Principe de l'élaboration des normes :

1. Les normes ISO répondent à un besoin du marché ;
2. Les normes ISO sont fondées sur une expertise mondiale ;
3. Les normes ISO sont le fruit d'un processus multipartite ;
4. Les normes ISO se fondent sur un consensus.

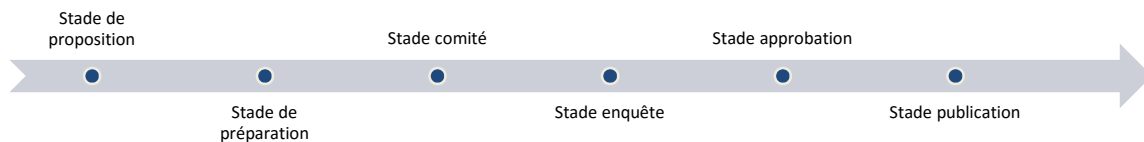


Figure 1.2 : Processus d'élaboration d'une norme

Tel que montré dans la Figure 1.2, le processus d'élaboration comporte 6 phases :

1. **Proposition** : La première étape de l'élaboration d'une norme internationale consiste à confirmer qu'il existe un besoin pour la norme internationale en question.
2. **Préparation** : En général, un groupe de travail est mis en place pour préparer un projet de travail. Plusieurs projets de travail successifs peuvent être examinés jusqu'à ce que le groupe de travail ait acquis la certitude d'avoir élaboré la meilleure solution technique au problème considéré.
3. **Comité** : La rédaction de la norme (ou autre référentiel) intervient à ce stade en suivant les règles de formatage ISO.
4. **Enquête** : Le projet de norme internationale est soumis à tous les comités membres de l'ISO pour vote et observations.
5. **Approbation** : Le projet final de norme internationale est soumis avec les observations formulées au stade enquête. Le texte est approuvé en tant que Norme internationale si une majorité des deux tiers des membres se prononce en sa faveur.
6. **Publication** : La norme est publiée.

Ce processus est long et il faut parfois attendre plusieurs années avant d'en arriver à l'étape de publication.

Chapitre 2 : L'utilisabilité

Dans ce chapitre, nous allons nous pencher sur la notion d'utilisabilité. Nous allons tenter de comprendre les concepts qui se cachent derrière cette caractéristique et nous finirons par présenter les normes ISO relatives.

Evidemment, dans ce chapitre, nous nous intéresserons à l'utilisabilité des systèmes/logiciels informatisés.

2.1 Utilité, utilisabilité et ergonomie

Dans les premières définitions (Shackel, 1981, 1986), le noyau conceptuel de l'utilisabilité est constitué de quatre composantes : l'efficacité, l'apprenabilité, la flexibilité du système et l'attitude de l'utilisateur envers le système.

Dans un des derniers raffinements de la définition, Shackel (1991) décrit l'utilisabilité d'un système comme « sa capacité, en termes fonctionnels humains, à permettre une utilisation facile et effective par une catégorie donnée d'utilisateurs, avec une formation et un support adapté, pour accomplir une catégorie donnée de tâches, à l'intérieur d'une catégorie spécifique de contextes ».

A cet égard, la définition donnée par les normes ISO et très proche de celle de Shackel (1991) : « Degré selon lequel un produit peut être utilisé, par des utilisateurs identifiés, pour atteindre des buts définis avec efficacité, efficience et satisfaction, dans un contexte d'utilisation spécifié » (ISO 9241-11, 1998). [CPFU 2003]

Les apports théoriques et méthodologiques concernant le concept d'utilisabilité sont nombreux. Cette notion possède de multiples acceptions.

L'usage d'un instrument (outil, logiciel, ...) peut être caractérisé selon deux axes :

- Son **utilité** :
C'est la capacité d'un objet à faciliter la réalisation d'une activité humaine.
- Son **utilisabilité** :
C'est la capacité d'un objet à être facilement utilisé par une personne donnée pour réaliser la tâche pour laquelle il a été conçu.

Les critères de l'utilisabilité sont :

- L'efficacité : le produit permet à ses utilisateurs d'atteindre le résultat prévu.
- L'efficience : le résultat est atteint avec un effort moindre ou requiert un temps minimal.
- La satisfaction : le confort et l'évaluation subjective de l'interaction pour l'utilisateur.

L'**ergonomie**¹, qui est une notion proche de celle d'utilisabilité, caractérise l'adaptation d'un système au travail et au bien-être des êtres humains.

L'utilisabilité et l'ergonomie concernent tout type d'instrument destiné à aider l'être humain dans son travail. Nous nous intéresserons, dans ce mémoire, à son application à l'outil informatique et plus particulièrement aux logiciels et aux sites web.

¹ Du grec *ergon* : travail et *nomos* : règle, loi naturelle.

2.2 Enjeux de l'utilisabilité

L'utilisabilité est un critère très important, voir décisif, sur plusieurs plans en fonction du contexte dans lequel on la mesure.

Citons les enjeux suivants :

- **Productivité**
L'utilisabilité influence la productivité d'un logiciel, cet aspect économique est un critère important pour le choix d'un logiciel utilisé à des fins professionnelles.
Un logiciel mal conçu, n'offrant pas une utilisabilité suffisante, nécessitant des périodes d'apprentissage ou des formations aura forcément un impact sur la productivité des utilisateurs.
- **Efficience**
Un logiciel efficient permettra de réaliser rapidement la tâche prévue, sans perte de temps et avec moins de stress. Ce qui contribue au maintien d'un climat de travail agréable. Un logiciel efficient aura un lien direct sur la productivité.
- **Succès**
L'utilisabilité conditionne très souvent la réussite commerciale d'un logiciel. C'est même fréquemment un critère plus important que la technicité, le coût ou l'esthétique.
Des études ont montré que les consommateurs sont plus attirés par l'esthétique ou la facilité de prise en main plutôt que par l'efficacité ou la qualité d'un produit.
- **Fidélisation**
C'est également un facteur de fidélisation : l'utilisateur achète les yeux fermés lorsqu'il est sûr de pouvoir utiliser facilement le logiciel.
Citons, comme exemple, les produits d'Apple qui misent, entre autres, sur son image de marque.
- **Réussite**
L'ergonomie de l'interface utilisateur constitue très souvent un facteur déterminant dans la réussite d'un projet informatique et son acceptation par les utilisateurs.
Ce critère passe fréquemment avant les aspects techniques, fonctionnels et de performance.

2.3 Heuristiques de Nielsen

Lors de la conception et du développement du système, il est important de sélectionner les propriétés requises et d'évaluer si le système conçu les remplit selon les objectifs fixés. Certains travaux fournissent des conseils et règles sur la manière de concevoir les systèmes interactifs, le plus largement cité étant celui de Jakob Nielsen, un des leaders dans le domaine de l'utilisabilité. Cependant, ces recommandations et standards ne fournissent pas de lignes directrices sur les moyens à mettre en œuvre.

Ci-dessous se trouve la liste¹ établie par Jakob Nielsen des heuristiques à suivre lorsque nous développons un logiciel :

1. **La visibilité de statut du système:** le système devrait toujours informer les utilisateurs de l'endroit où ils se trouvent et sur ce que fait le système.
Par exemple, dans le cas d'un téléchargement qui n'indique pas la progression du téléchargement du fichier, la personne pourrait cliquer plusieurs fois sur le bouton soumettre, empêchant sa requête d'aboutir ou téléchargeant le fichier plusieurs fois.
2. **Concordance entre le système et le monde réel :** le système doit pouvoir informer les utilisateurs dans un langage qu'ils comprennent.
Nous pouvons, par exemple, utiliser des métaphores afin de faire un lien entre le modèle mental du concepteur et celui des utilisateurs.
3. **Contrôle de l'utilisateur et liberté :** trop souvent, l'utilisateur choisit un menu ou clique sur un lien par erreur. Il doit pouvoir retourner facilement d'où il vient. Il doit pouvoir maîtriser les actions qu'il fait sur le système.
4. **Consistance et standards :** utiliser des standards et des normes graphiques, une iconographie et un vocabulaire normalisés.
Un utilisateur ne devrait pas se demander si différents mots ou boutons font la même chose.
5. **Prévenir les erreurs :** prévoir les erreurs que les utilisateurs pourraient faire de façon à les éviter. Dans la mesure du possible, il faut prévoir les erreurs et proposer des actions correctives.
6. **Reconnaissance plutôt que rappel :** les options, les menus et les actions devraient être visibles et repérables rapidement.
7. **Flexibilité et efficacité :** privilégier les raccourcis pour les utilisateurs plus expérimentés.
8. **Design et esthétique minimalistes :** éviter de surcharger l'interface d'information visuelle inutile. De même, les textes doivent aller à l'essentiel. Les premières impressions visuelles ne doivent pas être négligées car elles ont un gros impact sur la perception de l'utilisateur.
Une approche KISS² est souvent une bonne solution.
9. **Fonction d'aide pour les utilisateurs :** les messages d'erreurs doivent être clairs, précis et indiquer clairement le problème et la solution.
10. **Aide et documentation :** fournir de l'aide et de la documentation appropriée pour utiliser le système (même si dans un monde idéal un site devrait être assez simple d'utilisation pour ne pas en avoir besoin).

¹ 10 Usability Heuristics for User Interface Design by JAKOB NIELSEN on January 1, 1995

² KISS = Keep It Simple, Stupid !!!

2.4 Conception centrée sur l'utilisateur

La conception centrée sur l'utilisateur ou conception orientée utilisateur (UCD) est une philosophie et une démarche de conception où les besoins, les attentes et les caractéristiques propres des utilisateurs finaux sont pris en compte à chaque étape du processus de développement d'un produit.

La conception centrée sur l'utilisateur est principalement utilisée en conception informatique et s'appuie sur des critères d'ergonomie et d'utilisabilité. Cette démarche se distingue fortement d'autres démarches de conception en cherchant à adapter le produit (généralement l'interface utilisateur) à l'utilisateur final plutôt que de lui imposer un mode d'utilisation choisi par les concepteurs.

La démarche de conception centrée sur l'utilisateur repose sur l'idée que les utilisateurs finaux sont les mieux placés pour évaluer et utiliser le produit.

De ce fait, le développement d'un produit est a priori davantage guidé par les besoins et exigences des utilisateurs finaux, plutôt que par des possibilités techniques ou technologiques.

Toutefois, l'utilisateur final peut être entendu de deux manières :

- l'utilisateur réel, le plus susceptible d'utiliser le produit répondant à ses exigences et étant éventuellement déjà utilisateur d'une version précédente du produit ;
- l'utilisateur potentiel qui présente des exigences proches ou équivalentes, et que l'utilisation du produit pourrait intéresser.

La définition et le recueil des besoins, des attentes et des exigences applicables au produit doivent être issus d'une démarche rigoureuse dans le cadre d'une intervention ergonomique, d'une enquête utilisateur, d'un test utilisateur. Ces étapes peuvent être effectuées avec un produit existant ou un prototype.

2.5 Standards en relation avec l'utilisabilité

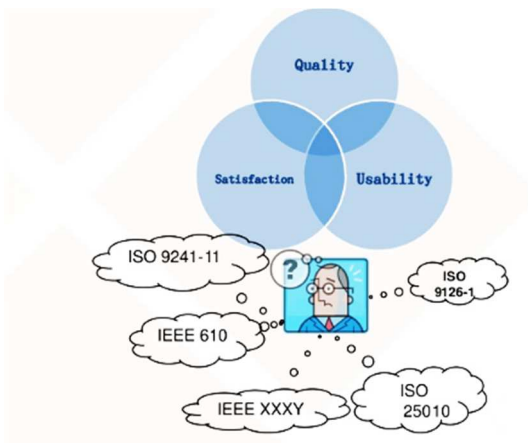


Figure 2.1 : Current Research - Usability Standards – XBOSOFT

Il existe un très grand nombre de standards ou de guidelines relatifs à l'utilisabilité.

Si nous ne regardons qu'au niveau des normes ISO, nous pouvons dénombrer près de 35 documents normatifs relatifs au thème de l'ergonomie et de l'utilisabilité.

L'objectif de ce chapitre n'est bien évidemment pas de présenter toutes les normes disponibles (rien que cette partie demanderait probablement un mémoire à part entière) mais plutôt de brosser brièvement une partie des documents publiés et en cours d'élaboration concernant l'ergonomie et l'utilisabilité des logiciels.

D'une manière générale, les années 1980 ont vu naître les premiers essais de définition de la notion d'utilisabilité¹ et l'idée d'une « conception centrée utilisateur » s'est développée graduellement.

Dans les premières définitions, le noyau conceptuel de l'utilisabilité était constitué de quatre composantes : l'efficacité, la facilité d'apprentissage, la flexibilité du système et l'attitude de l'utilisateur envers le système.

Dans un des derniers raffinements de la définition, Shackel (1991) décrit l'utilisabilité d'un système comme « sa capacité, en termes fonctionnels humains, à permettre une utilisation facile et effective par une catégorie donnée d'utilisateurs, avec une formation et un support adapté, pour accomplir une catégorie donnée de tâches, à l'intérieur d'une catégorie spécifique de contextes ». Cette définition met l'accent sur des mesures classiques de la performance en psychologie expérimentale : vitesse d'accomplissement de la tâche et taux d'erreurs.

De cette définition ressort l'idée que l'utilisabilité n'existe pas en tant que telle, mais qu'elle est le résultat de l'interaction entre un utilisateur, un dispositif (aussi simple soit-il), les tâches qu'il permet d'accomplir et un ou plusieurs contextes d'utilisation (utilisateur, système, tâche et contexte étant eux-mêmes des variables complexes). De ce fait, la question de la possibilité de définir à priori et une fois pour toutes, voire de normaliser et de certifier l'utilisabilité d'un système, reste entière. Néanmoins, il est encourageant de constater que l'ISO a commencé à inclure l'utilisabilité dans un certain nombre de normes et rapports techniques (nous pouvons, comme exemple, citer les quelques standards suivants : ISO 9241, ISO 13407, ISO 18529).

¹ Notamment dans les travaux de Shackel (1981, 1986), Eason (1984), Whiteside, Bennett et Holtzblatt (1988)

2.5.1 Classification

Durant les 30 dernières années, plusieurs standards en rapport avec l'interaction homme-machine (HCI) ont vu le jour. Certains donnent des lignes directrices et recommandations tandis que d'autres contiennent des spécifications bien précises.

Plus récemment, des experts ont travaillé à l'intégration de l'utilisabilité dans le génie logiciel et dans les normes de qualité du logiciel. Cela a suscité des compromis : par exemple, réconcilier les différentes définitions données, jusque-là, au concept de l'utilisabilité en adoptant le nouveau terme de « qualité à l'usage » pour le représenter.

Les standards relatifs à l'utilisabilité et l'HCI peuvent être classés en quatre catégories:

1. L'utilisation du produit
2. L'interface utilisateur et l'interaction
3. Le processus utilisé pour développer le produit
4. La capacité d'une organisation à appliquer la conception centrée utilisateur (UCD)

Si l'on s'intéresse à l'utilisabilité, il faudra s'inquiéter des différentes phases du développement logiciel. Aussi bien du produit logiciel lui-même que du processus de cycle de vie qui a permis de le concevoir. Et pour chacune de ses catégories, nous pouvons trouver des normes ISO relatives à l'utilisabilité.

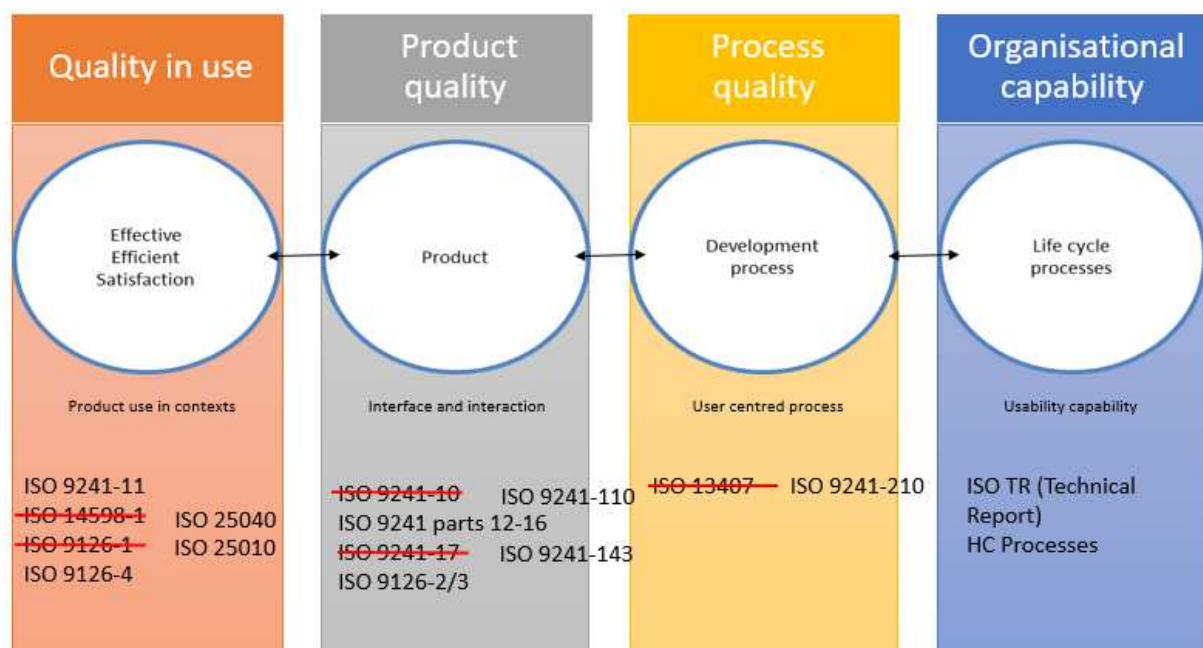


Figure 2.2 : Classement des standards relatifs à l'utilisabilité en catégories

Comme l'illustre la Figure 2.2, il y a des relations entre ces catégories. Pour que le logiciel développé réponde à des critères d'utilisabilité (Quality in use), il devra présenter des interfaces et interactions facile et ergonomique (Product quality). Pour ce faire, le processus de développement doit pouvoir intégrer une approche UCD, c'est-à-dire centré sur l'utilisateur, (Process quality) et ce processus doit s'intégrer dans le cycle de vie de l'organisation (Organisation capability).

La Figure 2.2 illustre les relations logiques entre ces catégories :

- **Quality in use :**
L'objectif est que le produit remplisse les caractéristiques de l'utilisabilité (à savoir : efficacité, efficacité et satisfaction) lorsqu'il est utilisé dans son contexte. A noter qu'il peut y avoir plusieurs contextes d'utilisation.
- **Product quality :**
Une condition préalable pour y parvenir est que l'interface et l'interaction soient appropriées.
- **Process quality :**
Cela nécessite un processus de conception centrée sur l'utilisateur
- **Usability capability :**
Le processus de développement doit s'inscrire dans un cycle de vie pouvant le supporter.

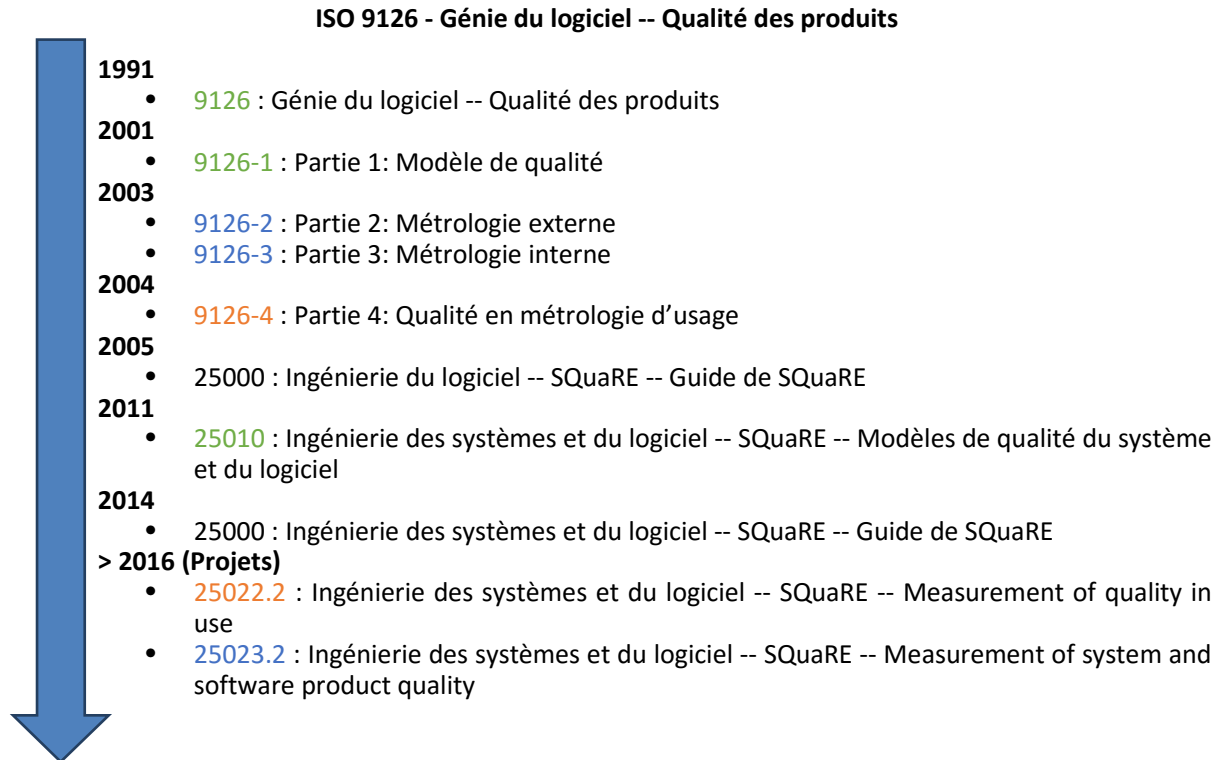
[HCI 2006]

Pour chaque catégorie, nous retrouvons donc plusieurs standards qui permettent d'améliorer le processus de développement logiciel en mettant l'accent sur les aspects d'utilisabilité, d'ergonomie et de conception centré sur l'utilisateur.

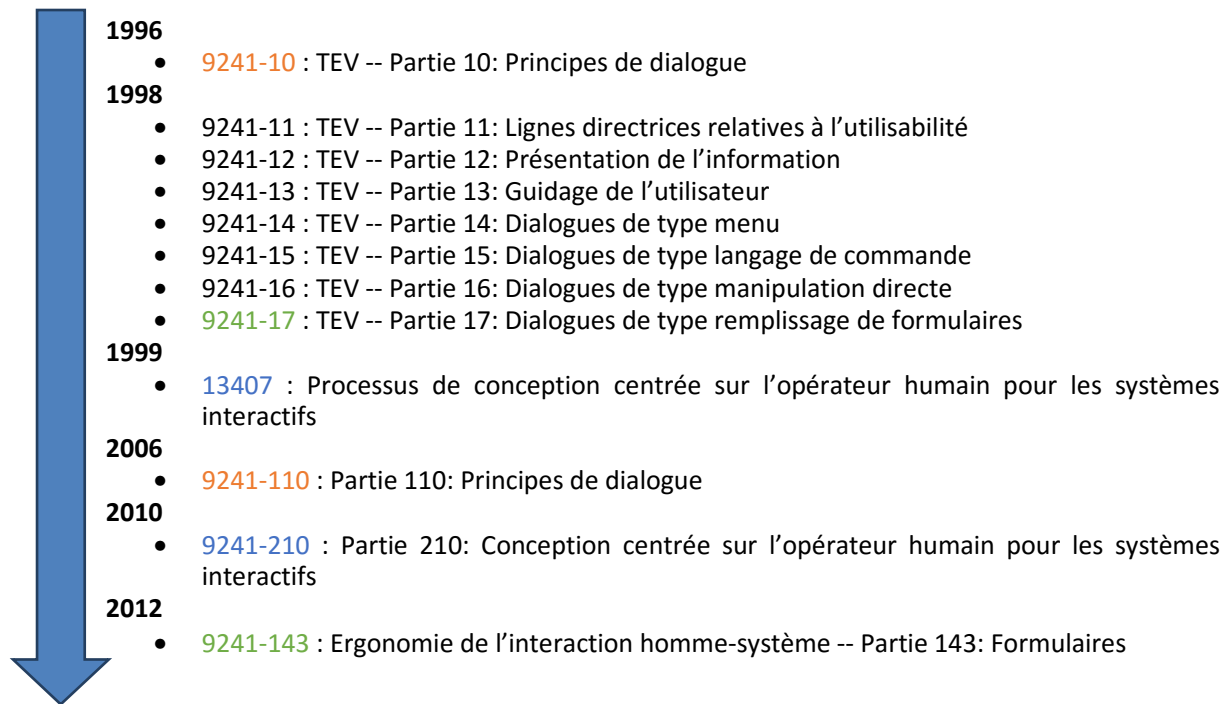
Nous constatons que la tendance actuelle est de remplacer ISO 9126 par ISO 25000 et de mettre à jour, progressivement, la norme ISO 9241.

2.5.2 Evolution des normes

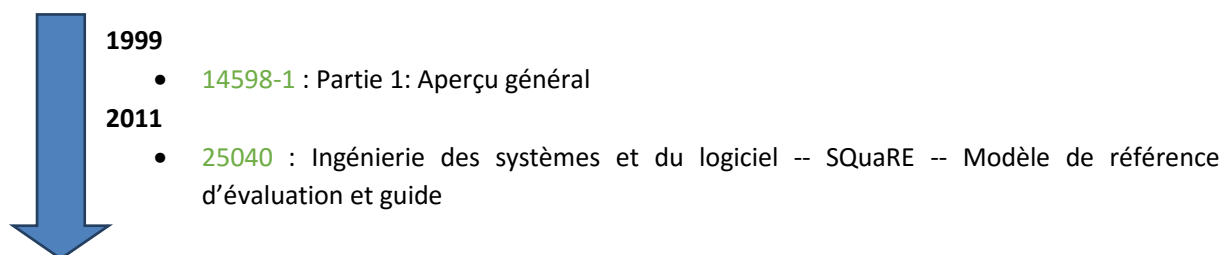
Afin de se rendre compte que certaines des normes citées dans ce mémoire ne sont pas récentes et que, par conséquent, elles ont été revues ou remplacées, nous allons voir comment les normes ISO 9126, ISO 9241 et ISO 14598 ont évolué.



Comme nous pouvons le constater, SQuaRE, qui est apparu en 2005, est le successeur d'ISO 9126.

ISO 9241 - Ergonomie de l'interaction homme-système

Actuellement, ISO 9241 est toujours en cours de révision et de nouvelles parties continues d'être publiées.

ISO 14598 - Technologies de l'information -- Évaluation de produits logiciels

SQuaRE remplace donc la norme ISO 14598.

Chapitre 3 : ISO 9241 – Ergonomie de l'interaction homme-système

Maintenant que nous avons vu le concept d'utilisabilité, nous allons nous pencher sur la norme ISO qui concerne l'ergonomie de l'interaction entre l'homme et le système, c'est-à-dire ISO 9241 et ses différentes parties.

3.1 Organisation de la norme

ISO 9241 a été publié en 1992 et comprend différentes parties traitant toutes de l'ergonomie de l'interaction entre l'homme et le système.

Sur la Figure 3.1, nous retrouvons la structure de la norme tel que décrit dans sa partie 1¹ :

- Partie 1: Introduction générale
- Partie 2: Guide général concernant les exigences des tâches
- Partie 3: Exigences relatives aux écrans de visualisation
- Partie 4: Exigences relatives aux claviers
- Partie 5: Exigences relatives à l'aménagement du poste de travail et aux postures
- Partie 6: Exigences relatives à l'environnement
- Partie 7: Exigences d'affichage concernant les réflexions
- Partie 8: Exigences relatives aux couleurs affichées
- Partie 9: Exigences relatives aux dispositifs d'entrée autres que les claviers
- Partie 10: Principes de dialogue
- Partie 11: Lignes directrices concernant l'utilisabilité
- Partie 12: Présentation de l'information
- Partie 13: Lignes directrices pour l'utilisateur
- Partie 14: Dialogues de type menu
- Partie 15: Dialogues de type langage de commande
- Partie 16: Dialogues de type manipulation directe
- Partie 17: Dialogues de type remplissage de formulaires

¹ ISO 9241-1:1997 Exigences ergonomiques pour travail de bureau avec terminaux à écrans de visualisation (TEV) -- Partie 1: Introduction générale

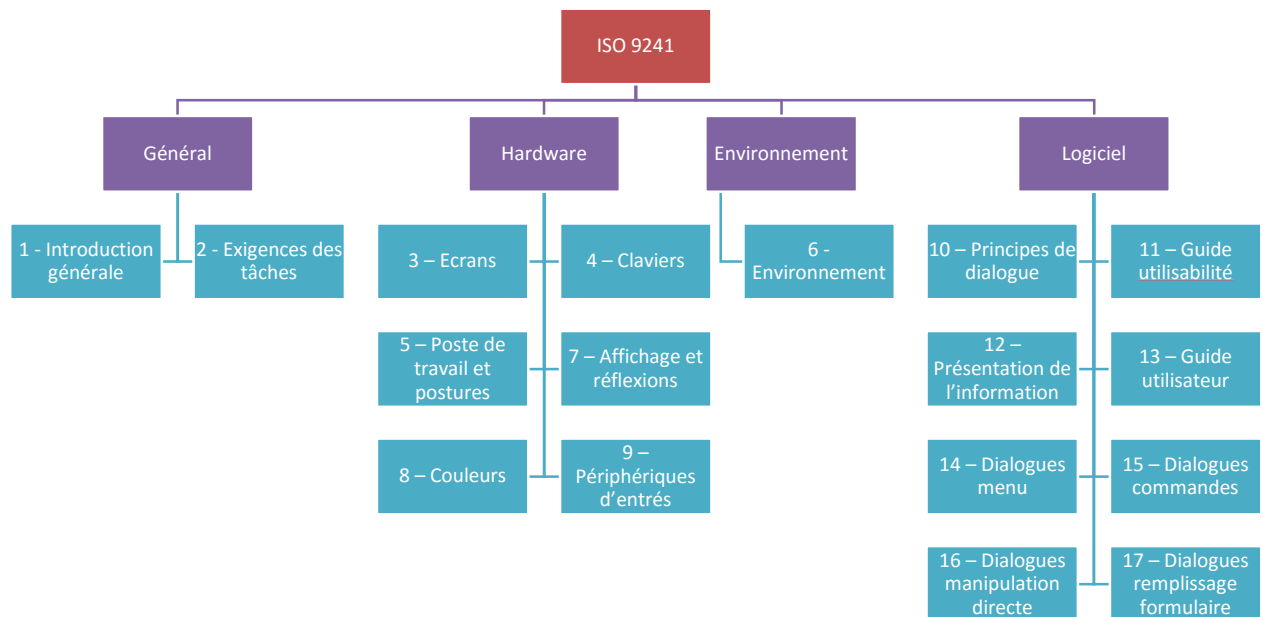


Figure 3.1 : Organisation de la norme ISO 9241

Dans ce mémoire, nous nous intéresserons aux parties « Logiciel » de la norme.

Il est à noter que depuis 1997 (date de la dernière version d'ISO 9241-Partie 1: Introduction générale), la norme a évolué. Certaines parties ont été revues, renommées et ajoutées :

- Partie 20: Lignes directrices sur l'accessibilité de l'équipement et des services des technologies de l'information et de la communication (TIC)
- Partie 100: Introduction aux normes relatives à l'ergonomie des logiciels
- Partie 110: Principes de dialogue
- Partie 151: Lignes directrices relatives aux interfaces utilisateurs Web
- Partie 171: Lignes directrices relatives à l'accessibilité aux logiciels
- Partie 210: Conception centrée sur l'opérateur humain pour les systèmes interactifs
- Partie 300: Introduction aux exigences relatives aux écrans de visualisation électroniques
- Partie 302: Terminologie relative aux écrans de visualisation électroniques
- Partie 303: Exigences relatives aux écrans de visualisation électroniques
- Partie 304: Méthodes d'essai de la performance de l'utilisateur pour écrans de visualisation électroniques
- Partie 305: Méthodes d'essai de laboratoire optique pour écrans de visualisation électroniques
- Partie 306: Méthodes d'appréciation sur le terrain des écrans de visualisation électroniques
- Partie 307: Méthodes d'essai d'analyse et de conformité pour écrans de visualisation électroniques
- Partie 308: Écrans à émission d'électrons par conduction de surface (SED) [Rapport technique]
- Partie 309: Écrans à diodes électroluminescentes organiques (OLED) [Rapport technique]
- Partie 400: Principes et exigences pour les dispositifs d'entrée physiques
- Partie 410: Critères de conception des dispositifs d'entrée physiques
- Partie 420: Procédures de sélection pour les dispositifs d'entrée physiques
- Partie 910: Cadre pour les interactions tactiles et haptiques
- Partie 920: Lignes directrices relatives aux interactions tactiles et haptiques

3.2 Lien entre les parties 11, 110 et 12

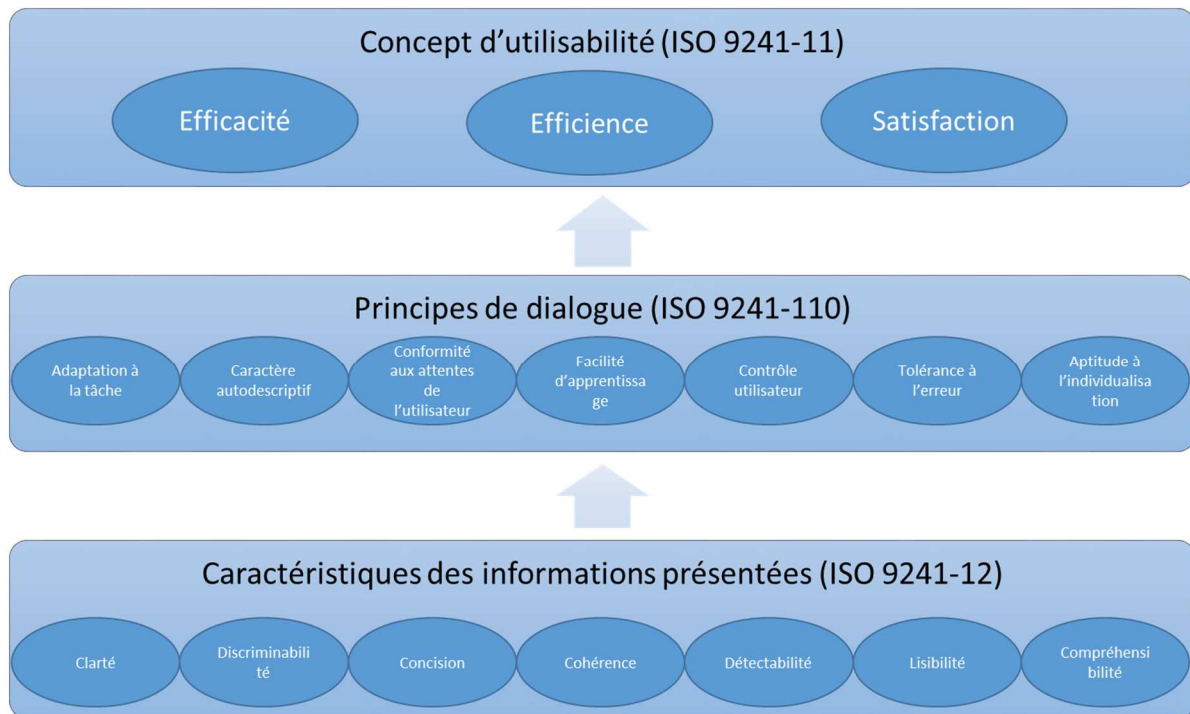


Figure 3.2 : Lien entre 9241-11, 9241-110 et 9241-12

Comme l'illustre la Figure 3.2, nous pouvons voir que la partie 12 de la norme est plutôt « bas niveau » et concerne la façon dont on présente les informations.

La partie 110, quant à elle, s'intéresse aux dialogues entre le système et l'utilisateur. Comme nous le verrons plus tard (Chapitre 5 : Croisement des normes), toutes les caractéristiques présentées sont reprises, et parfois enrichies, dans ISO 25000.

La partie 11, qui concerne l'utilisabilité, est un peu plus abstraite et haut niveau. Nous verrons également qu'ISO 25000 s'est fortement inspiré de cette partie pour établir son modèle de « Qualité à l'usage ».

3.3 Focus sur les parties d'ISO 9241

Bien que toutes les parties de la norme soient intéressantes, nous allons nous intéresser aux parties suivantes :

- ISO 9241-11:1998 - Lignes directrices relatives à l'utilisabilité

Cette partie explique que pour spécifier ou mesurer l'utilisabilité, il est nécessaire d'identifier les objectifs et de décomposer l'efficacité, l'efficience, la satisfaction et le contexte d'utilisation en sous-composants comportant des attributs qui peuvent être mesurés et vérifiés.

Il n'existe qu'une seule version de cette partie datant de 1998. Une révision est à l'état de projet et vise, notamment, à introduire le concept d'expérience utilisateur dans la notion d'utilisabilité.

L'utilisabilité est définie par la norme ISO 9241-11 comme
« Le degré selon lequel un produit peut être utilisé, par des utilisateurs identifiés, pour atteindre des buts définis avec **efficacité**, **efficience** et **satisfaction**, dans un contexte d'utilisation spécifié ».

[ISO/IEC 9241-11]

- ISO 9241-110:2006 - Principes de dialogue

Cette partie remplace ISO 9241-10:1996. Elle énonce des principes de conception ergonomiques formulés en termes généraux (c'est-à-dire qu'ils sont présentés sans référence à des situations d'utilisation, à des applications, à des environnements ou à une technologie) et fournit un cadre relatif à l'application de ces principes dans l'analyse, la conception et l'évaluation des systèmes interactifs.

Elle est centrée sur les principes de dialogue associés à la conception ergonomique du dialogue entre l'utilisateur et le système interactif, et ne tient pas compte de tout autre aspect de conception tel que le marketing, l'esthétique et la conception interne à l'entreprise.

Cette partie est destinée aux concepteurs d'outils de développement d'interfaces utilisateurs, de guides de style et d'interfaces utilisateurs, aux développeurs, aux acheteurs et aux évaluateurs.

- ISO 9241-210 (2010) - Conception centrée sur l'opérateur humain pour les systèmes interactifs

Cette partie remplace ISO 13407:1999 - Processus de conception centrée sur l'opérateur humain pour les systèmes interactifs.

Elle fournit des exigences et des recommandations relatives aux principes et aux activités de conception centrée sur l'opérateur humain, intervenant tout au long du cycle de vie des systèmes informatiques interactifs. Elle est destinée à être utilisée par les responsables de la gestion des processus de conception, et traite des manières dont les composants matériels et les logiciels des systèmes interactifs permettent d'améliorer l'interaction homme-système.

Chapitre 4 : ISO 25000 - Exigences de qualité des systèmes et du logiciel et évaluation

Dans ce chapitre, nous allons nous pencher sur ISO 25000, aussi connu sous le nom de SQuaRE.

Cette norme est assez récente¹ car elle a été créée en 2005 et les différentes parties sont encore, actuellement, en cours de rédaction.

Comme le montre la Figure 4.1, ce standard est amené à remplacer la norme ISO 9126 et ISO 14598. Cette norme a pour objectif de poser le cadre et la référence pour définir les exigences qualité d'un logiciel (ISO 9126) et la manière dont ces exigences seront évaluées (ISO 14598).



Figure 4.1 : SQuaRE = ISO 9126 + ISO 14598

Nous allons présenter, brièvement, ces deux normes (ISO 9126 et ISO 14598) afin d'avoir une idée sur leur contenu et nous présenterons ensuite SQuaRE.

4.1 ISO 9126 - Technologies de l'Information : Qualités des produits logiciels

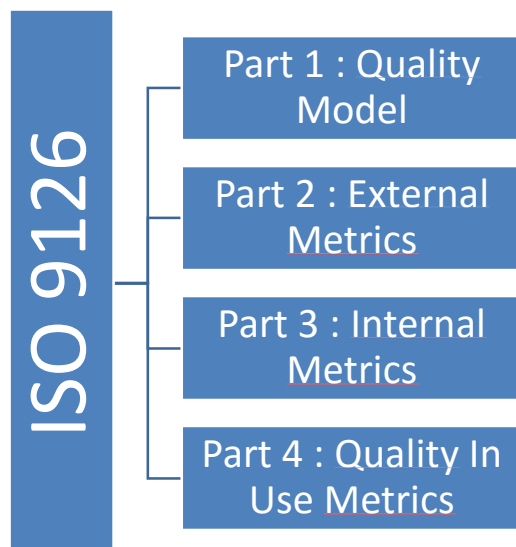


Figure 4.2 : Organisation de la norme ISO 9126

La norme ISO 9126 définit un ensemble d'indicateurs pour la qualité logicielle (Figure 4.3 : ISO 9126 - Caractéristiques internes et externes d'un produit), et « facilite » ainsi le processus d'évaluation logiciel et la spécification d'exigences fonctionnelles ou non-fonctionnelles. Cette norme est en application depuis 1991.

Un des avantages de ce modèle est qu'il identifie les caractéristiques de qualité interne/externe du produit logiciel ainsi que la qualité à l'usage. En effet, la distinction entre la qualité interne et la qualité externe est très importante car l'une va s'intéresser aux artefacts statiques du produit (code, spécifications, analyses, ...) tandis que l'autre va s'intéresser à la qualité du produit pendant son exécution.

Toutefois, il a l'inconvénient de ne pas faire apparaître clairement comment ces aspects peuvent être mesurés.

[ORTEGA, PÉREZ & ROJAS 2003]

¹ Même si elle a plus de 10 ans, les différentes parties ont été principalement publiées après 2010 ou sont encore à l'état de projet. C'est pour cette raison que nous pouvons la considérer comme récente.

Cette norme pose les caractéristiques qualité principales d'un produit logiciel et organise leur classification afin d'en faciliter son utilisation lors de processus d'évaluation logiciel.

Celle-ci a été réalisée par le comité technique JTC 1 de l'ISO/IEC. Les évolutions de cette norme ont été intégrées dans la norme SQuaRE (que nous verrons, ci-dessous, dans le point 0), ou ISO 25000.

Les caractéristiques décrites s'appliquent au logiciel (progiciels, développements spécifiques, firmware...) que l'on doit considérer comme un système de transformation avec des données en entrée et des données en sortie.

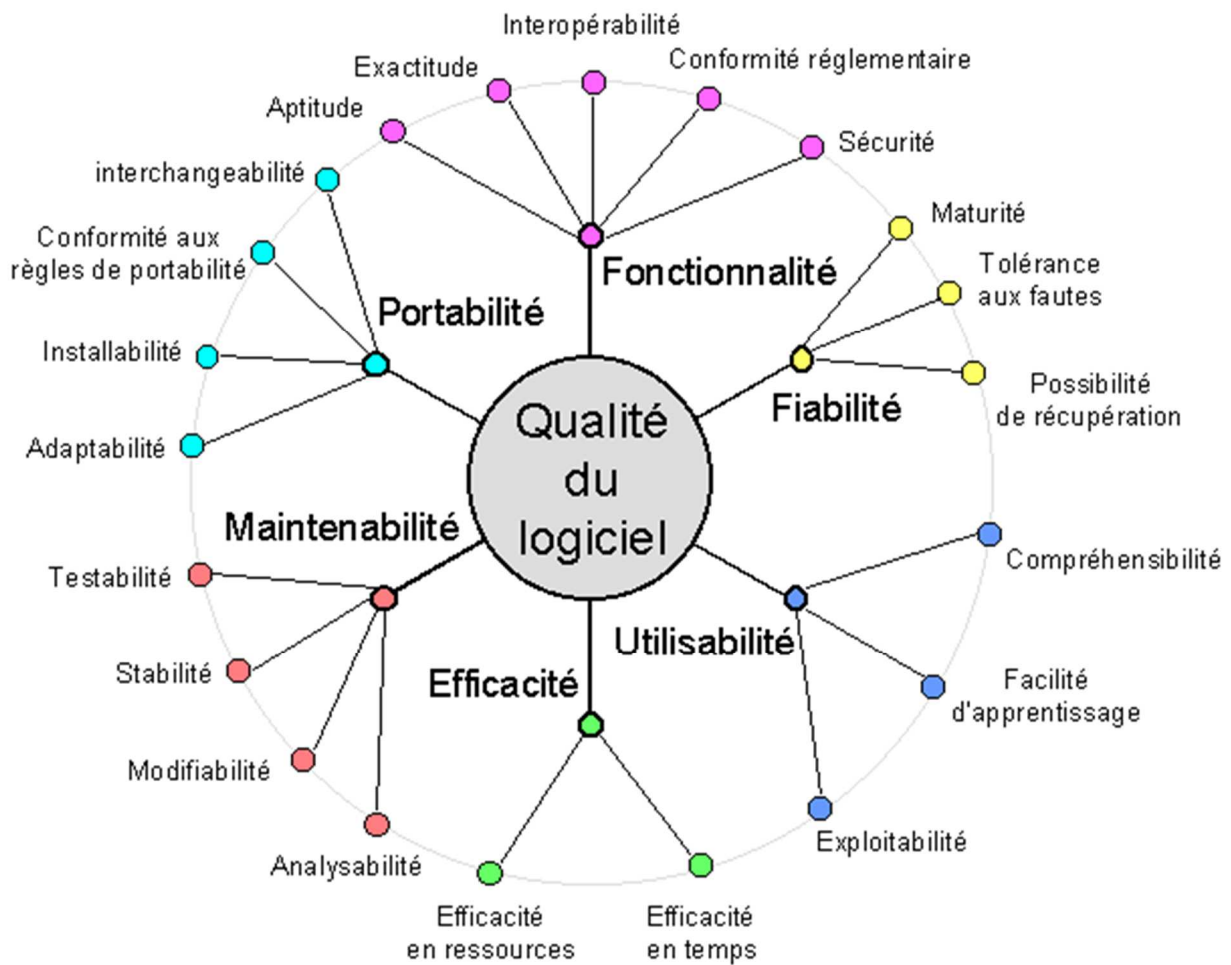


Figure 4.3 : ISO 9126 - Caractéristiques internes et externes d'un produit

Les caractéristiques qualité définies par cette norme sont :

- La capacité fonctionnelle
 - L'aptitude
 - L'exactitude
 - L'interopérabilité
 - La conformité réglementaire
 - La sécurité
- La fiabilité
 - La maturité
 - La tolérance aux fautes
 - La capacité de récupération
- La facilité d'usage (utilisabilité)
 - L'exploitabilité
 - La facilité d'apprentissage
 - La facilité de compréhension
- L'efficacité
 - L'efficacité des ressources employées
 - L'efficacité des temps de réalisation
- La maintenabilité
 - La stabilité
 - La facilité de modification
 - La facilité d'analyse
 - La facilité à être testé
- La portabilité
 - La facilité d'installation
 - La facilité de migration
 - L'adaptabilité
 - L'interchangeabilité

Ce classement a été défini en 2001. L'évolution de technologies logicielles et la puissance des traitements ont depuis lors modifié une partie de ce classement. Par exemple, la stabilité d'un logiciel est une caractéristique qui influe directement sur l'exactitude et la précision et on la classerait plutôt dans la capacité fonctionnelle. L'interopérabilité devient, de plus en plus, une « super » caractéristique applicable à toutes les fonctions. L'interopérabilité serait donc mieux gérée comme caractéristique principale et dont l'une des sous-caractéristiques serait la capacité fonctionnelle.

L'ordre de ce classement est important car il reflète, en partie, les enchaînements de causes à effets entre les caractéristiques. Cette influence est déterminante dans la construction du système d'indicateurs pour le pilotage de la performance d'un logiciel.

4.2 ISO 14598 - Technologies de l'information : Évaluation du produit logiciel

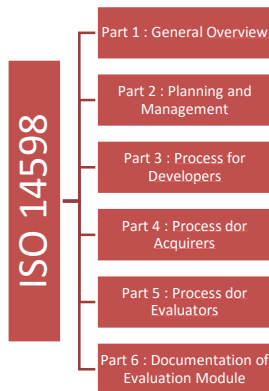


Figure 4.4 : Organisation de la norme ISO 14598

La norme ISO/IEC définit la méthodologie d'évaluation de produit logiciel afin de mesurer la conformité du produit avec les exigences. L'analyse des résultats d'évaluation peut aussi être utilisée dans un contexte d'amélioration des processus.

Cette norme comporte six parties et fournit un processus d'évaluation selon trois perspectives : le développeur, l'acquéreur et l'évaluateur.

La partie cinq, qui s'applique dans une perspective d'évaluateur, décrit le processus d'évaluation étape par étape afin de permettre l'expression des besoins d'évaluation en termes de caractéristiques de qualité tels que définis dans la norme ISO/IEC 9126.

- 14598-1 : Évaluation de produits logiciels – Aperçu général
- 14598-2 : Évaluation du produit – Planification et gestion
- 14598-3 : Évaluation de produits logiciels – Procédés pour développeurs
- 14598-4 : Évaluation de produits logiciels – Procédés pour acquéreurs
- 14598-5 : Évaluation de produits logiciels – Procédés pour les évaluateurs
- 14598-6 : Évaluation de produits logiciels – Documentation des modules d'évaluation

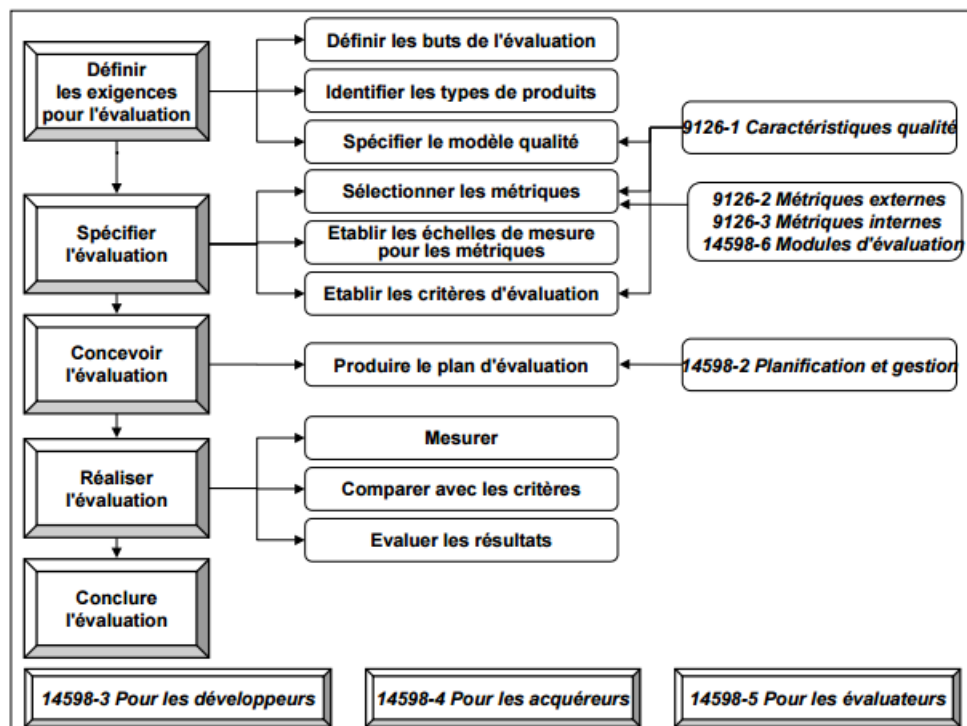


Figure 4.5 : L'évaluation du produit logiciel s'appuie sur les normes

4.3 SQuaRE

ISO 25000, aussi connue sous le nom de SQUARE, crée un cadre de travail commun pour évaluer la qualité du produit logiciel.

Ce standard définit les exigences qualité et la mise en œuvre de leur évaluation pour un produit logiciel. Cette norme reprend, enrichit et remplace les normes ISO 9126 et ISO 14598 citées précédemment.

Le SEI a édité des recommandations pour une utilisation intégrée avec la qualité produit comme objectif de la performance des processus. Concrètement, ceci se traduit par la mise en correspondance de certains processus du CMMI avec ceux du SQuaRE (ISO 25000).

Dans le référentiel CMMI, les caractéristiques du produit sont abordées dans les processus de « Technical Solution » et de « Product Integration ».

Ensuite, le CMMI procède à la vérification du produit fini avec le processus de « Project Management », de « Verification » et de « Validation ».

Les modèles de qualité peuvent s'inspirer des étapes de la qualité présentées par l'ISO 9126 et intégrées dans le SQuaRE, telles que :

- la qualité interne,
- la qualité externe,
- la qualité à l'utilisation.

Les mesures de qualité relient les exigences aux critères d'acceptation.

Ainsi, l'ISO 25000 apporte une aide précieuse pour la réalisation du processus de « Requirement Development », de « Verification » et de « Validation » du CMMI.

4.4 Organisation de la norme

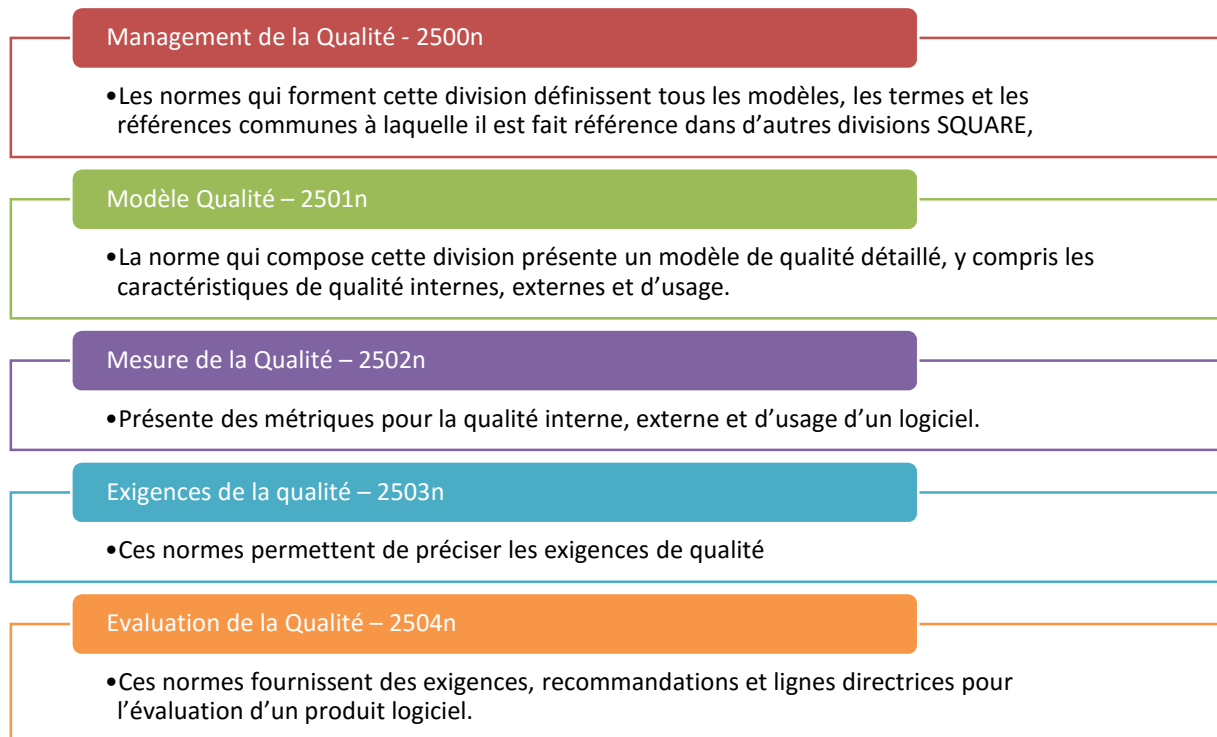


Figure 4.6 : Organisation de la norme ISO 25000

4.4.1 ISO/IEC 2500n – Quality Management Division

- 25000:2014 – Guide to Square
- 25001:2014 – Planning and Management

4.4.2 ISO/IEC 2501n – Quality Model Division

- 25010:2011 – Quality model
- 25012:2008 – Data quality model

4.4.3 ISO/IEC 2502n – Quality Measurement Division

- 25020:2007 – Measurement reference model and guide
- 25021:2012 – Quality measure elements
- 25022 – Measurement of quality in use
[En cours de publication]
- 25023 – Measurement of system and software product quality
[En cours de publication]
- 25024:2015 – Measurement of data quality

4.4.4 ISO/IEC 2503n- Quality Requirement Division

- 25030:2007 – Quality requirements

4.4.5 ISO/IEC 2504n – Qualityt Evaluation Division

- 25040:2011 – Evaluation process
- 25041:2012 – Evaluation guide for developers, acquirers and independent evaluators
- 25045:2010 – Evaluation modules for recoverability

4.4.6 ISO/IEC 25050 to 25099 – SQuaRE extensions

- 25051:2014 – Requirement for quality of Ready to Use Software Product (RUSP) and instructions for testing
- 25060:2010 – Common Industry Format (CIF) for usability : General framework for usability-related information
- 25062:2006 - Common Industry Format (CIF) for usability test reports
- 25063:2014 - Common Industry Format (CIF) for usability : Context of use description
- 25064:2013 - Common Industry Format (CIF) for usability : User needs report
- 25065 - Common Industry Format (CIF) for usability : User Requirements Specification
[Nouveau projet approuvé]
- 25066 - Common Industry Format (CIF) for usability : Usability Evaluation Report
[En cours de publication]

4.5 Modèles de qualité

Les modèles les plus connus actuellement (ISO 9126, SQuaRE, McCall, GQM) sont des modèles hiérarchiques qui recensent les principes de qualité en partant des exigences globales et des principes les plus généraux pour descendre vers les métriques qui permettent de les mesurer. Ceci implique que la mesure de la qualité ne peut débuter qu'une fois le modèle totalement spécifié et que les premiers résultats ne peuvent être obtenus qu'une fois la collecte des données suffisante.

Ces modèles définissent un langage commun pour modéliser les qualités d'un logiciel. Le langage de description utilise des termes tels que « facteurs qualité », « caractéristiques », « sous-caractéristiques » et « métriques » pour classer de façon arborescente et structurée, sur la base de définitions standardisées, un vocabulaire de plusieurs dizaines de propriétés.

Le système SQuaRE décrit deux modèles distincts. Un modèle de qualité lié à l'utilisation du logiciel et un modèle de qualité propre à la production logicielle.

4.5.1 Qualité à l'Usage¹

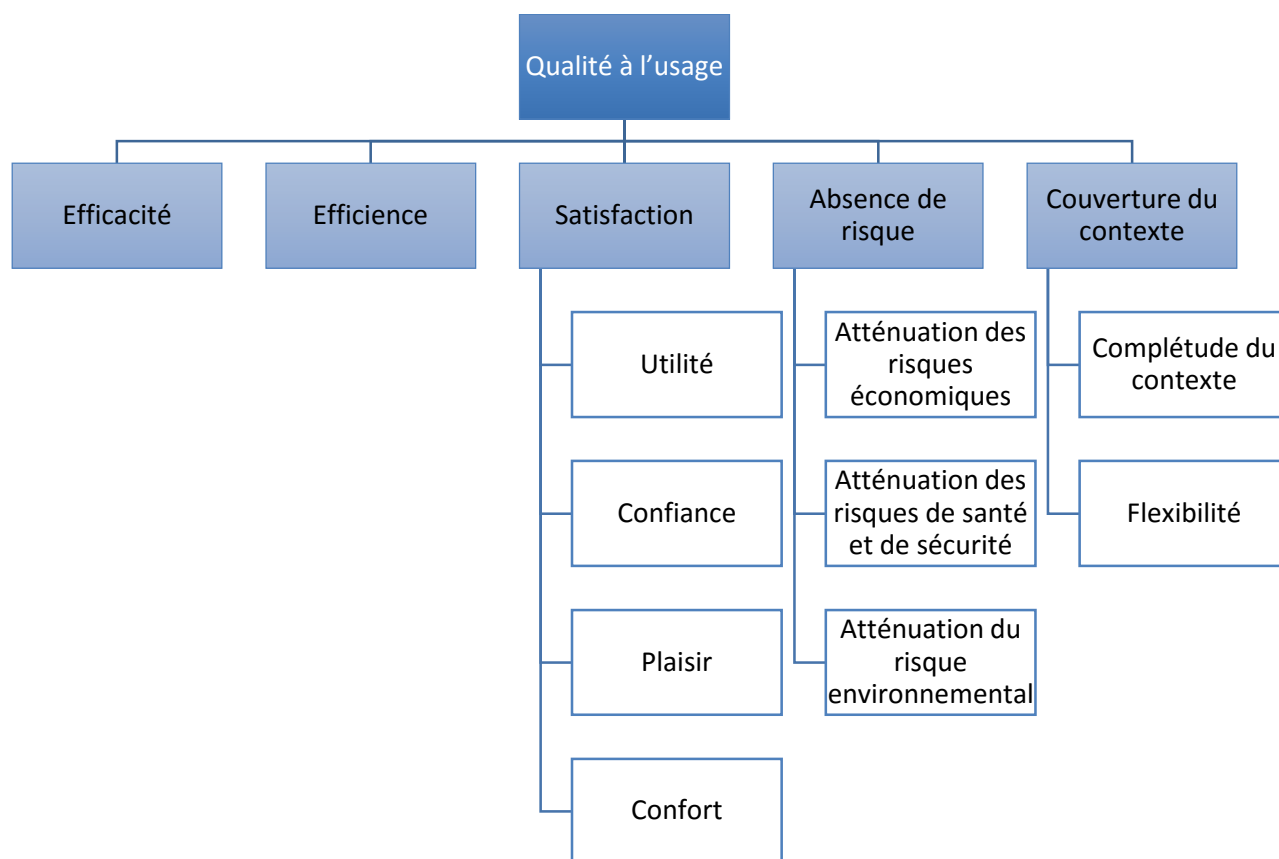


Figure 4.7 : ISO 25000 - Modèle Qualité à l'usage

¹ ISO 25000 n'a pas encore été traduit en français. Ce modèle est une traduction proposée dont l'original se trouve dans l'Annexe 1 : SQuaRE - Quality In Use

4.5.2 Qualité Produit¹

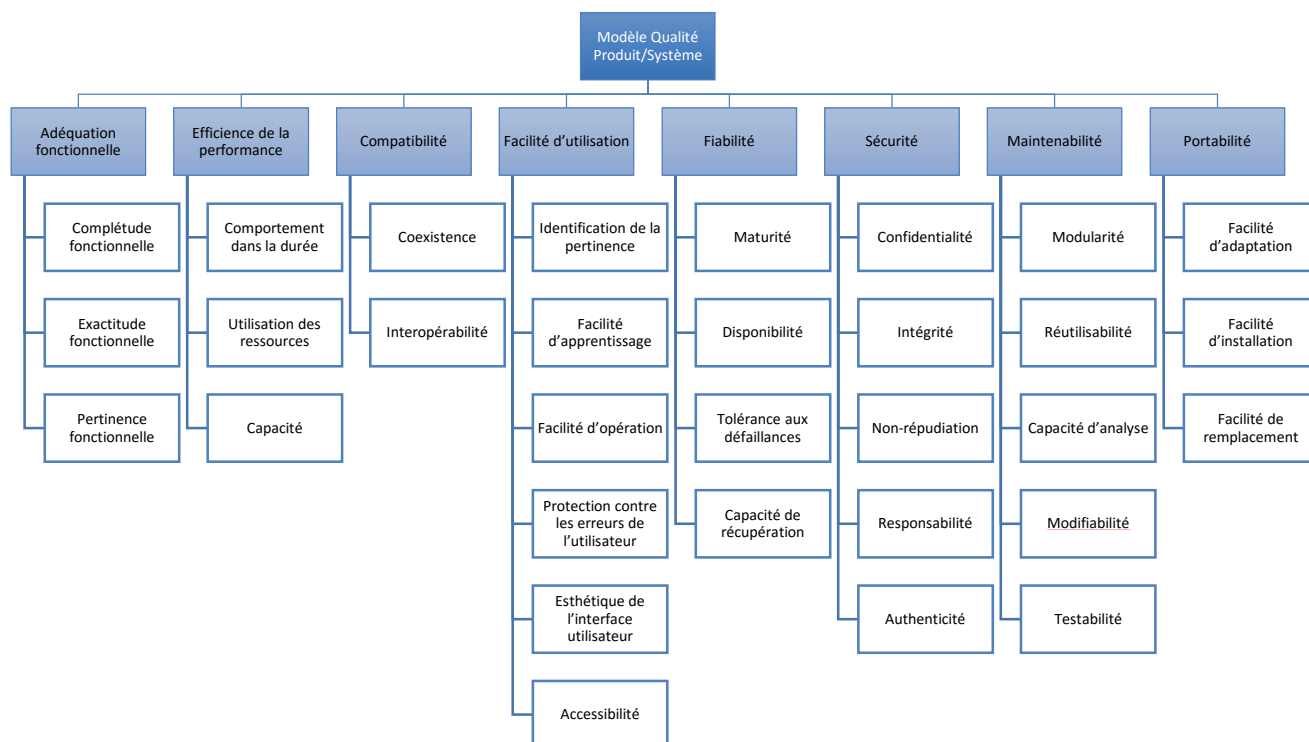


Figure 4.8 : ISO 25000 - Modèle Qualité Produit

4.6 Caractéristiques concernées par l'utilisabilité

Il est évident que toutes les caractéristiques présentées dans ISO 25010 valent la peine d'être prises en compte lors de l'évaluation d'un produit ou d'un système. Cependant, dans le cadre de ce mémoire, nous allons nous intéresser aux caractéristiques ayant un impact² sur l'utilisabilité. Nous verrons également si toutes les parties de SQuaRE nécessaires à cette étude sont déjà publiées.

Dans le modèle « Qualité à l'usage », nous nous intéresserons aux caractéristiques suivantes: efficacité, efficience et Satisfaction.

Dans le modèle « Qualité produit », nous nous intéresserons principalement aux caractéristiques suivantes : adéquation fonctionnelle, facilité d'utilisation, portabilité.

¹ ISO 25000 n'a pas encore été traduit en français. Ce modèle est une traduction proposée dont l'original se trouve dans l'Annexe 2 : SQuaRE - System/Software Product Quality

² Le choix des caractéristiques d'ISO 25000 liés à l'utilisabilité sera pris en croisant les différentes normes citées ou en faisant une hypothèse. Il est donc discutable que d'autres caractéristiques, non prises en compte, puissent avoir une influence plus ou moins forte sur l'utilisabilité.

Chapitre 5 : Croisement des normes

Jusqu'ici, nous avons pu voir ce qu'était une norme et l'intérêt de s'y référer lors de la mise en place d'un processus tel que le cycle de vie logiciel.

Nous avons également vu le concept d'utilisabilité et d'ergonomie appliquée au développement logiciel qui est une étude spécifique des relations entre l'humain et l'environnement logiciel qu'il utilise.

Concernant l'ergonomie, nous avons brossé ce que propose l'Organisation Internationale de Normalisation du point de vue de l'utilisabilité et nous nous sommes penchés sur la norme ISO 9241 qui est axée sur l'ergonomie de l'interaction homme-système et la conception centrée utilisateur qui est une approche, parmi d'autres, fréquemment utilisée qui permet de guider la conception humain-machine (IHM) en prenant en considération les caractéristiques des utilisateurs, et de se focaliser sur l'utilisabilité du produit conçu.

Nous avons pu constater que cette norme n'est pas récente. A titre d'exemple, il suffit de prendre en considération la partie 11¹ qui donne les lignes directrices concernant l'utilisabilité pour s'en rendre compte. En effet, celle-ci a été publiée en 1998 et depuis, nous devons nous poser la question d'adéquation de cette norme avec l'évolution des systèmes tels que nous les connaissons actuellement. Depuis 1998, les logiciels ont beaucoup évolué en terme d'ergonomie et d'interaction homme-système. Notamment avec l'arrivée des nouvelles technologies tactiles pour le grand public.

Nous avons également vu la norme ISO 25000 qui, elle, met l'accent sur la qualité logicielle et définit des caractéristiques et sous-caractéristiques permettant de mesurer la qualité d'un logiciel. Le guide SQuaRE est plus récent qu'ISO 9241 car la plupart des différentes parties ont été publiées après 2010. De plus, SQuaRE est encore, actuellement, en train de travailler sur d'autres parties qui sont à l'état de projet et ne seront publiées que plus tard.

Au niveau des études ou recherches menées dans la cadre de l'utilisabilité, nous pouvons trouver beaucoup de sources parlant indépendamment des deux mais, actuellement, aucune ne fait le lien entre ces deux normes.

Les recherches de Bevan N., en 1997, font principalement référence à la norme ISO 9126, mais l'âge de l'article, antérieur à la préparation de la norme ISO 25000, fait que l'auteur s'attache plus à décrire les différentes caractéristiques d'utilisabilité sans les mettre en regard avec l'évolution du domaine de l'interaction homme-machine, ce qui est le but de ce mémoire. [QUALITY&USABILITY]

Du fait que SQuaRE est une norme assez récente, le type d'approche proposée dans ce mémoire est une première. En effet, les documents scientifiques se préoccupent souvent d'une norme bien précise et non d'établir des liens et mettre en rapport plusieurs normes. Ce mémoire amène donc une première revue, centrée sur l'utilisabilité, en tentant d'établir les liens et de soulever les incohérences entre cette dernière et ISO 9241.

Dans ce chapitre, nous allons nous intéresser aux liens entre la norme ISO 9241 et ISO 25000. Certains seront assez logiques et très peu discutables et d'autres liens feront l'objet d'une revue critique.

Nous allons parcourir les parties 9241-11 et 9241-110 afin de mapper les caractéristiques avec celles de SQuaRE.

¹ Exigences ergonomiques pour travail de bureau avec terminaux à écrans de visualisation (TEV) – Partie 11 : Lignes directrices concernant l'utilisabilité, NBN EN ISO 9241-11, 1^e éd., juillet 1998

5.1 Lien entre ISO 9241-11:1998 et ISO 25010:2011

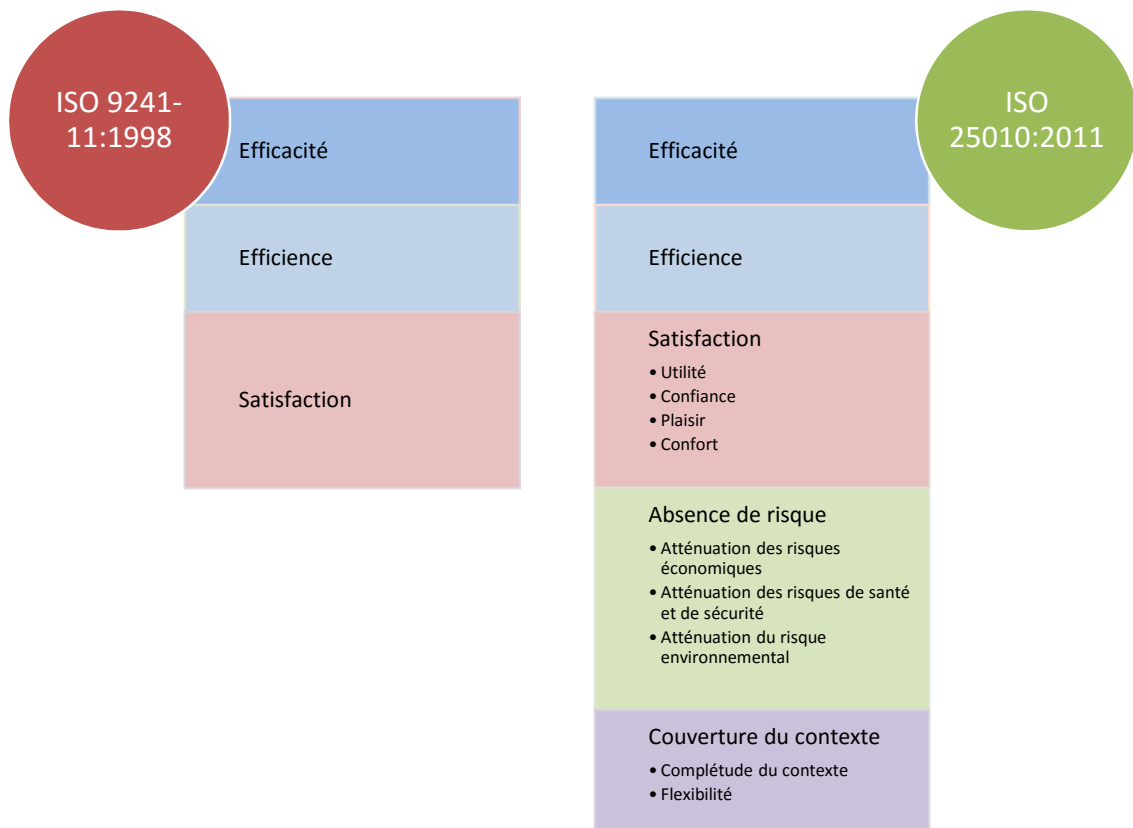


Figure 5.1 : Caractéristique d'ISO 9241-11 et le modèle à l'usage d'ISO 25010

Nous allons parcourir les caractéristiques en partant de celles définies dans ISO 9241 et en les mettant en regard des caractéristiques d'ISO 25010. Nous verrons, ainsi, s'il existe un lien entre les caractéristiques des deux normes et si SQuaRE a tenu compte d'ISO 9241 lors de l'écriture de la norme.

5.1.1 Efficacité

Efficacité - « Précision et degré d'achèvement selon lesquels l'utilisateur atteint des objectifs spécifiés. »

[ISO/IEC 9241-11], [ISO/IEC 25000]

La définition qu'ISO 9241 avait donné, en 1998, à la caractéristique d'efficacité est toujours valable actuellement. ISO 25010 l'a reprise telle qu'elle était présentée dans la norme.

Nous caractériserons donc un système/logiciel d'efficace s'il permet à l'utilisateur de mener à bien la tâche qu'il entreprend.

5.1.2 Efficience

Efficience - « Rapport entre les ressources dépensées et la précision et le degré d'achèvement selon lesquels l'utilisateur atteint des objectifs spécifiés. »
[ISO/IEC 9241-11], [ISO/IEC 25000]

Comme pour l'efficacité, ISO 25010 se base de nouveau sur ISO 9241 et la référence dans le cadre de l'efficience.

Le système/logiciel sera efficace s'il permet à l'utilisateur de mener à bien une tâche précise dans un temps satisfaisant. Encore faudra-t-il se mettre d'accord sur la notion de temps satisfaisant afin de répondre au mieux aux besoins des utilisateurs du produit.

5.1.3 Satisfaction

Satisfaction - « Absence d'inconfort, et attitudes positives dans l'utilisation du produit. »
[ISO/IEC 9241-11]

Satisfaction

« Degré avec lequel les besoins utilisateurs sont satisfaits lorsque le système ou le produit est utilisé dans un contexte d'utilisation spécifique. »

Utilité - « L'utilisateur est satisfait de l'accomplissement de ses objectifs. Ceci inclut les résultats et conséquences de son utilisation. »

Confiance - « L'utilisateur ou les intervenants ont confiance que le produit ou le système se comporte comme attendu. »

Plaisir - « L'utilisateur apprécie l'utilisation du produit ou du système lors de l'accomplissement de ses tâches. »

Confort - « L'utilisateur est satisfait du confort physique. »
[ISO/IEC 25000]

Ici, nous remarquons qu'ISO 25010 a enrichi la caractéristique de satisfaction. En effet, il ajoute le concept d'utilité et introduit ceux liés à l'expérience utilisateur. Dans ISO 9241, la partie 210 aborde la conception centrée sur l'opérateur humain pour les systèmes interactifs mais celle-ci se concentre sur le processus et non sur des caractéristiques. La définition que donne ISO 25010 est donc plus précise et nous nous rendons mieux compte de ce que l'on entend par satisfaction.

ISO 25010 introduit également la notion de contexte d'utilisation dans la satisfaction. L'utilisation que l'on a du produit doit être réalisée dans un contexte spécifique afin que la notion de satisfaction ait un sens.

5.2 Lien entre ISO 9241-110:2006 et ISO 25010:2011



Figure 5.2 : Caractéristique d'ISO 9241-110 et ISO 25010

Tout comme dans le chapitre précédent, nous allons également parcourir les caractéristiques en partant de celles définies dans ISO 9241 et en les mettant en regard des caractéristiques d'ISO 25010. Nous verrons, ainsi, s'il existe un lien entre les caractéristiques des deux normes et si SQuaRE a tenu compte d'ISO 9241 lors de l'écriture de la norme.

5.2.1 Adaptation à la tâche

« Un système interactif est adapté à la tâche lorsqu'il permet à l'utilisateur de réaliser cette tâche, c'est-à-dire lorsque les fonctionnalités et le dialogue sont fondés sur les caractéristiques de la tâche (plutôt que sur la technologie choisie pour accomplir cette tâche). »

[ISO/IEC 9241-110]

Adéquation fonctionnelle

« Le produit fournit des fonctions qui répondent aux besoins exprimés et implicites lorsque le produit est utilisé dans des conditions spécifiées. »

Complétude - *« L'ensemble des fonctions couvre toutes les tâches spécifiées et les objectifs de l'utilisateur. »*

Exactitude - *« Le produit donne des résultats corrects avec le degré de précision nécessaire (ni trop, ni trop peu). »*

Pertinence - *« Les fonctions sont adaptées pour des tâches déterminées et les objectifs de l'utilisateur. »*

[ISO/IEC 25000]

Dans ISO 25010, c'est la sous-caractéristique « Pertinence » qui référence l'adaptation à la tâche. Cependant, comme ISO 25010 le fait, nous pouvons dire qu'un système est adapté à la tâche s'il permet d'effectuer cette tâche de manière complète, avec exactitude et pertinence. Les sous-caractéristiques que nous retrouvons dans ISO 25010 sont donc justifiées et viennent compléter la définition donnée par ISO 9241.

5.2.2 Aptitude à l'individualisation

« Un dialogue permet l'individualisation lorsque les utilisateurs peuvent modifier l'interaction et la présentation de l'information pour les adapter à leurs capacités personnelles et à leurs besoins. »

[ISO/IEC 9241-110]

Couverture du contexte

Flexibilité – *« Un produit ou système peut être utilisé avec efficacité, efficience, absence de risque et satisfaction au-delà du contexte initialement prévu »*

Portabilité

Facilité d'adaptation – *« Un produit ou un système peut être, de façon efficace et efficiente, adapté pour différents changements ou évolutions ou être adapté en fonction de l'individualisation. »*

[ISO/IEC 25000]

Nous allons retrouver la notion d'individualisation dans les deux modèles proposés par ISO 25010. Dans le modèle de qualité à l'usage, la sous-caractéristique « flexibilité » y fait référence. La flexibilité d'un système/logiciel permet de voir dans quelle mesure celui-ci peut être utilisé par des types d'utilisateurs

différents de ceux prévus initialement pour atteindre des objectifs additionnels. C'est la capacité que le système/logiciel propose pour s'adapter à de nouveaux utilisateurs, tâches et environnements.

Dans le modèle qualité du produit, nous allons également retrouver, sous l'attribut « Portabilité », la « facilité d'adaptation ». Cette sous-caractéristique englobe aussi cette notion d'individualisation.

Dans ce cas-ci, la notion d'individualisation est dispersée entre les deux modèles proposés par SQuaRE. Si nous nous concentrons uniquement sur l'utilisabilité, nous pourrions nous demander si une sous-caractéristique entièrement consacrée à l'individualisation n'aurait pas eu sa place dans le modèle de qualité à l'usage présenté dans ISO 25010. Cependant, le rôle de SQuaRE est aussi de donner des éléments de mesure de toutes ses caractéristiques et sous-caractéristiques et c'est probablement pour cette raison que ce choix a été pris.

Nous pouvons également constater qu'ISO 25010 élargit le scope de l'aptitude à l'individualisation par rapport à celui donné dans ISO 9241. En effet, dans ISO 9241, nous n'avons qu'une seule approche qui est que l'utilisateur peut adapter le système afin que ce dernier réponde à ses besoins. Tandis que dans ISO 25010, en plus de l'approche proposée dans ISO 9241, nous retrouvons une autre approche qui est que le logiciel/système a la faculté de s'adapter automatiquement à l'utilisateur sans intervention de celui-ci.

5.2.3 Facilité d'apprentissage

« Un dialogue permet l'apprentissage lorsqu'il soutient et guide l'utilisateur dans l'apprentissage de l'utilisation du système. »

[ISO/IEC 9241-110]

Facilité d'utilisation

« Degré selon lequel un produit ou système peut être utilisé par les utilisateurs spécifiés pour atteindre leurs objectifs avec efficacité, efficience, absence de risque et satisfaction dans le contexte d'utilisation. »

Facilité d'apprentissage - *« Le produit/système peut être utilisé par les utilisateurs spécifiés et les guide dans l'apprentissage de l'utilisation avec efficacité, efficience, absence de risque et satisfaction. »*

[ISO/IEC 25000]

Concernant la facilité d'apprentissage, ISO 25010 reprend les notions d'ISO 9241 mais les complète avec les notions d'« utilisateurs spécifiés »¹, « efficacité, efficience et satisfaction » et « absence de risque »². Nous retrouvons également la notion de contexte d'utilisation.

ISO 25010 a donc enrichi la notion de facilité qui avait été définie dans ISO 9241. En effet, si l'apprentissage est facile, nous devons également considérer les questions d'efficacité, d'efficience, d'absence de risque et de satisfaction.

¹ Cela signifie que l'on ne peut juger la facilité d'apprentissage que si nous nous adressons au public ciblé par le logiciel/système.

² Absence de risque que nous retrouvons dans le modèle qualité à l'usage (cf. 4.5.1 Qualité à l'Usage)

5.2.4 Caractère autodescriptif

« Un dialogue est autodescriptif dans la mesure où, à tout moment, il apparaît toujours clairement aux utilisateurs dans quel dialogue ils se situent, où ils se situent à l'intérieur de ce dialogue, quelles actions peuvent être entreprises et comment elles peuvent être exécutées. »

[ISO/IEC 9241-110]

Facilité d'utilisation

Facilité d'opération - *« Le produit/système fonctionne simplement et est facilement contrôlable. »*

Facilité d'apprentissage - *« Le produit/système peut être utilisé par les utilisateurs spécifiés et les guide dans l'apprentissage de l'utilisation avec efficacité, efficience, absence de risque et satisfaction. »*

[ISO/IEC 25000]

Nous pouvons retrouver cette notion de « caractère autodescriptif » derrière deux sous-caractéristiques dans ISO 25010 : La facilité d'opération et la facilité d'apprentissage.

ISO 9241 définit beaucoup plus clairement ce que l'on entend lorsque l'on parle d'autodescriptif. Nous ne retrouvons pas celle-ci de manière explicite. En regroupant cette notion sous la facilité d'opération, il y a un risque que la caractéristique d'auto-description d'une tâche soit perdue si on ne regarde que du côté de SQuaRE. Cette caractéristique est pourtant très importante car, comme nous l'avons vu au point « 2.3 - Heuristiques de Nielsen », la première heuristique énoncée par Nielsen est « la visibilité de statut du système » qui correspond à cette notion de caractère autodescriptif.

5.2.5 Contrôle utilisateur

« Un dialogue est contrôlable lorsque l'utilisateur peut initier et contrôler la direction et le rythme de l'interaction jusqu'à ce que le but ait été atteint. »

[ISO/IEC 9241-110]

Facilité d'utilisation

Facilité d'opération - *« Le produit/système fonctionne simplement et est facilement contrôlable. »*

[ISO/IEC 25000]

Pour cette caractéristique, nous remarquons qu'ISO 25010 a réduit la définition que donnait ISO 9241. En effet, ISO 9241 donne une définition un peu plus précise de ce que l'on entend par « contrôle utilisateur ». ISO 25010 donne, comme définition, que le système/logiciel doit être « facilement contrôlable » mais ne donne pas plus d'indication sur ce que l'on entend.

Nous pouvons penser que le choix pris par SQuaRE pour cette caractéristique vient de la difficulté de lui donner une mesure précise. En effet, il ne faut pas perdre de vue qu'à terme, le but est de pouvoir mesurer et s'assurer que les caractéristiques sur lesquelles nous voulons nous concentrer sont atteintes.

Cette notion s'éclaircira probablement lorsqu'ISO 25022 et ISO 25023 seront publiés.

5.2.6 Conformité aux attentes de l'utilisateur

« Un dialogue est conforme aux attentes de l'utilisateur s'il correspond aux besoins contextuels prévisibles de l'utilisateur et aux conventions communément admises. »
[ISO/IEC 9241-110]

Facilité d'utilisation

Facilité d'opération – « Le produit/système fonctionne simplement et est facilement contrôlable. »
[ISO/IEC 25000]

La sous-caractéristique « Facilité d'opération » va englober plusieurs caractéristiques d'ISO 9241-110. C'est une facilité qu'a pris ISO 25010 afin de prendre en compte toutes les caractéristiques que définit la partie 110 de la norme ISO 9241. Ici, nous voyons que la notion de conformité est réduite¹. Il faudra faire référence à ISO 9241 afin de voir ce que l'on entend par les « conventions communément admises ».

5.2.7 Tolérance à l'erreur

« Un dialogue est tolérant aux erreurs si, malgré des erreurs de saisie évidentes, le résultat prévu peut être obtenu soit sans action corrective, soit avec une action corrective minimale de la part de l'utilisateur. La tolérance à l'erreur est réalisée au moyen du contrôle des erreurs (contrôle des conséquences dommageables), de la correction des erreurs ou de la gestion des erreurs pour remédier aux erreurs qui surviennent. »
[ISO/IEC 9241-110]

Facilité d'utilisation

Protection contre les erreurs de l'utilisateur – « Le système protège les utilisateurs contre les erreurs qu'ils pourraient faire. »

Facilité d'opération – « Le produit/système fonctionne simplement et est facilement contrôlable. »
[ISO/IEC 25010]

ISO 9241 donne une définition très précise de ce que l'on entend par « tolérance à l'erreur ». ISO 25010 ne va pas aussi loin dans la définition qu'il donne. Il invite, d'ailleurs, à se référer à la définition d'ISO 9241.

¹ Si nous nous en tenons seulement à la sous-caractéristique « Facilité d'opération » d'ISO 25010. D'autres caractéristiques, tel que l'« Adéquation fonctionnelle », peuvent venir compléter cette notion afin de retrouver tous les aspects qu'ISO 9241-110 définit.

5.3 Mapping entre ISO 9241 et ISO 25000

Comme nous venons de le voir à travers ce chapitre, des liens existent entre la norme ISO 9241 et la norme ISO 25000. Certains ne sont pas discutables car ils sont directement référencés dans la norme et ISO 25000 nous invite à consulter ISO 9241 pour plus de précision sur la caractéristique en question. Par contre, comme nous avons pu le constater, d'autres liens sont plus sujets à interprétation ou à discussion.

La démarche suivie dans ce chapitre afin d'établir des liens entre ces deux normes a été, dans un premier temps, de prendre connaissance des caractéristiques proposées par chacune d'entre elles.

Ensuite, sur base des références proposées et d'hypothèses sur lesquelles nous nous sommes basés pour établir les liens, nous avons tenté d'établir un modèle retraçant les liens entre les deux normes.

Comme le montre la « Figure 5.3 : Mapping entre ISO 9241 et ISO 25000 », nous sommes partis des sous-parties 11 (Concept d'utilisabilité) et 110 (Principes de dialogues) d'ISO 9241. Ces parties définissent, elles aussi, des caractéristiques que nous avons tenté de mettre en regard de celles définies dans SQuaRE.

Concernant ISO 9241-11, nous constatons que les caractéristiques sont liées aux caractéristiques du modèle de qualité à l'usage que définit ISO 25010. Les liens établis sont des liens 1-1.

Concernant ISO 9241-110, nous avons procédé de la même manière et nous avons pu lier ses caractéristiques au modèle de qualité produit défini dans ISO 25010. Pour cette partie, ce sont les sous-caractéristiques d'ISO 25010 qui sont liées à ISO 9241-110. Nous constatons, également, que les liens établis ne sont pas tous 1-1 dans ce cas-ci.

Ce modèle a l'avantage de montrer une vue simplifiée des liens entre les deux normes mais il permet aussi de facilement se reporter à une caractéristique d'ISO 25010 à partir d'ISO 9241 et vice versa.

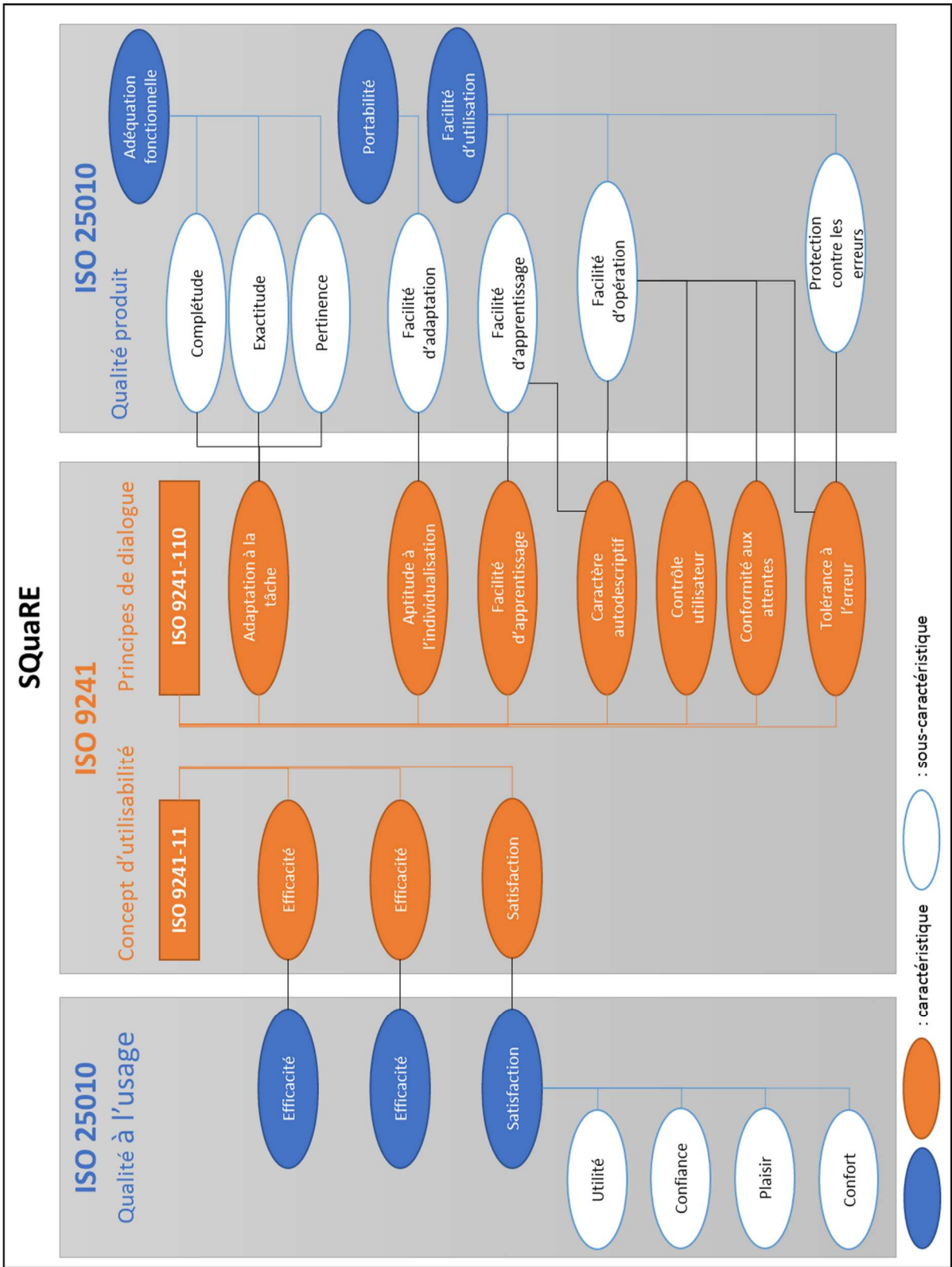


Figure 5.3 : Mapping entre ISO 9241 et ISO 25000

5.4 Les caractéristiques d'ISO 25010

Comme nous venons de le voir, SQaRE a tenu compte de la norme ISO 9241 pour toutes les caractéristiques relatives à la notion d'utilisabilité. Certaines caractéristiques sont enrichies et donnent une définition plus précise et/ou complète tandis que d'autres ont été rassemblées dans une caractéristique plus globale.

L'avantage des caractéristiques d'ISO 25010 est qu'elles permettent de cibler un ensemble d'exigences non-fonctionnelles sur lesquels nous voulons nous concentrer lors de la conception d'un produit. Cela permet d'avoir une base commune entre les modèles mentaux des utilisateurs et des concepteurs et de mettre un nom sur des caractéristiques qu'il est parfois difficile d'identifier.

Grâce à ces caractéristiques, il sera donc plus facile de discuter avec les utilisateurs afin de bien comprendre quels sont leurs besoins non-fonctionnels. Néanmoins, si l'on opte pour ce genre d'approche, il faudra probablement faire un travail de « vulgarisation »¹ de certaines caractéristiques car, malgré les définitions données par SQaRE, certaines risqueraient de ne pas être bien interprétées ou de couvrir un champ trop large (par exemple, nous avons vu que beaucoup de concepts se cachent derrière la facilité d'opérabilité). En effet, une critique qui revient beaucoup lorsque l'on parle de norme est qu'elles sont souvent très abstraites et loin de la réalité du terrain.

5.5 Cas pratique

Afin d'introduire le chapitre suivant, consacré aux mesures, nous avons mené une petite expérience en environnement réel afin de valider les concepts abordés dans ce chapitre.

L'« Annexe 3 : Rapport sur la qualité d'un produit » montre un exemple d'utilisation dans un contexte réel. Il s'agit d'une méthode de mesure facile à mettre en place et basée sur ISO 25010. Après explication, il a été demandé aux utilisateurs² de donner un score³ allant de 0 à 3 sur un produit développé et ce, pour chaque sous-caractéristique relative à l'utilisabilité de la norme.

Cela a permis d'obtenir simplement des premières mesures concernant le produit développé. Du point de vue de l'utilisabilité, cet exercice a tout son sens puisque l'utilisabilité doit être jugée par les utilisateurs du logiciel et il a l'avantage d'être facile et très simple à mettre en place.

Par contre, les résultats obtenus ne sont que de simples appréciations et ne permettent pas de se rendre réellement compte de la qualité du produit. Pour ce faire, il faudra effectuer un travail un peu plus approfondi pour comprendre et interpréter les résultats que l'on a obtenu.

C'est pour cette raison que SQaRE, en plus de fournir des caractéristiques, va fournir des méthodes de mesure afin de s'assurer que le produit rencontre bien les exigences définies.

Le processus de relevé de métriques, d'interprétation et de mise en place d'actions correctives est un sujet très vaste qui mériterait probablement un mémoire à lui seul. Nous ne nous intéresseront, dans ce mémoire, qu'à une partie des méthodes de mesure proposées.

¹ Par vulgariser, nous voulons dire qu'il faut simplifier au maximum les définitions afin qu'elles puissent être comprises par n'importe quel utilisateur.

² Trois personnes ont participé à cette campagne de tests.

³ L'échantillonnage de ce score est un exemple et ne se base sur aucune étude. Il est donc, légitimement, discutable.

Chapitre 6 : Mesurer l'utilisabilité

« You can't control what you can't measure. »

Tom DeMarco

Une des situations les plus souvent citée lorsque l'on parle de mesure est celle de Tom DeMarco : « Vous ne pouvez pas maîtriser ce que vous ne mesurez pas ». Sans doute que cette affirmation est vraie, même s'il nous semble plus correct de nuancer et de dire que la mesure facilite la maîtrise. Ce qui est par contre certain, c'est que quand les mesures sont employées sans un cadre théorique clair, non seulement elles sont inutiles mais elles peuvent créer de fausses assurances qui, dans le travail technique, peuvent facilement conduire à des catastrophes.

Nous avons vu, jusqu'ici, quelles sont les caractéristiques proposées par SQuaRE lorsque nous nous intéressons à l'utilisabilité d'un système/logiciel. Ces caractéristiques de qualité et les mesures associées sont très utiles pour évaluer un logiciel mais aussi pour définir les exigences de qualité. La standardisation des caractéristiques permet d'uniformiser les définitions afin que les différents intervenants parlent le même langage.

En fonction du contexte de développement, l'accent sera mis différemment sur les diverses caractéristiques. Il est important de cibler les caractéristiques pertinentes pour le système que l'on développe. Par exemple, un logiciel pour la commande de sandwich en interne dans une petite entreprise n'aura pas les mêmes besoins et moyens que le site Tax-on-web¹.

Ainsi, pour chaque caractéristique et sous-caractéristique définie dans le modèle de Qualité, nous allons déterminer les besoins des utilisateurs en utilisant un poids. A titre d'exemple, la Figure 6.1 montre comment nous pouvons déterminer les besoins en utilisant cette méthode.

Quality In Use		
	CHARACTERISTIC	WEIGHT (High/Medium/Low)
	Effectiveness	H
	Productivity	H
	Safety	L
	Satisfaction	M

Figure 6.1 : User Needs Characteristics & Weights

Si l'on veut pouvoir améliorer la qualité logicielle, encore faut-il être capable de mesurer cette qualité. Pour cela, il faut prendre en compte bien plus de facteurs que la mesure de la qualité du code source. Des mesures devront donc être prises afin de s'assurer que nous rencontrons bien les exigences de qualité souhaitées. C'est donc, logiquement, que nous allons voir ce que ISO 2502n (cf. Figure 6.2) propose comme méthode de mesure.

¹ Tax-on-web est un site web permettant aux citoyens belges de rentrer leur déclaration fiscale.
www.taxonweb.be

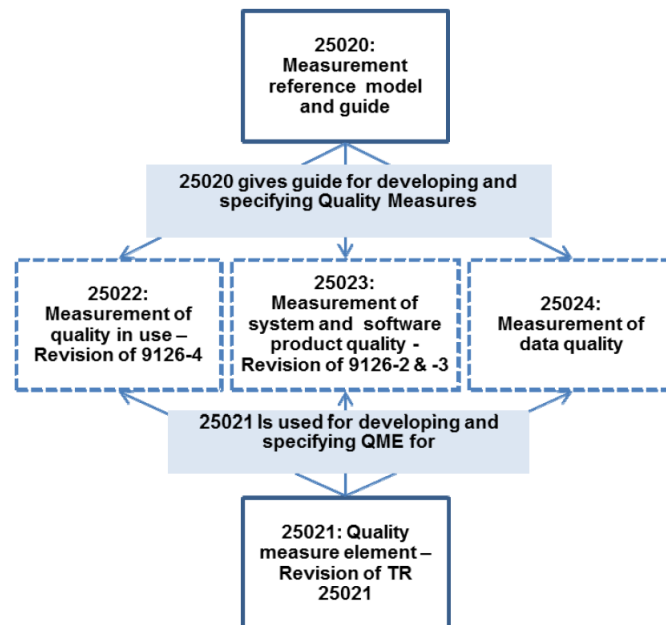


Figure 6.2 : Structure of the quality Measurement Division

ISO 25022 et 25023 sont les deux normes qui auraient dû nous intéresser dans ce chapitre. Or, comme nous l'avons vu dans le chapitre 4.4.3, ISO/IEC 2502n – Quality Measurement Division, ces deux normes sont toujours en cours de développement et ne sont pas encore disponibles. Nous allons donc devoir nous référer aux parties 2, 3 et 4 d'ISO 9126.

6.1 Métriques concernant l'utilisabilité

Les métriques d'utilisabilité mesurent le degré avec lequel le logiciel peut être compris, appris, exploité, attrayant et conforme.

La plupart des mesures d'utilisabilité seront testées par des utilisateurs qui essaieront d'utiliser une fonctionnalité du système. Les résultats seront influencés par la capacité des utilisateurs et la complexité du système qu'ils utilisent. Pour minimiser l'impact sur les mesures, le logiciel évalué sera exécuté dans des conditions spécifiées de façon explicite par un échantillon d'utilisateurs représentant un groupe d'utilisateurs identifiés.

Cela vaut d'ailleurs pour n'importe quelle caractéristique de qualité. Comme le souligne ISO 9126-2, l'échantillon d'utilisateurs choisi est très important pour évaluer les qualités externes d'un produit.

Comme l'illustre la Figure 6.3, nous pouvons voir que « l'espace de complexité » d'un système est défini par la dimension de la complexité de l'interaction et la complexité du domaine. La complexité de l'interaction, qui est représentée par l'axe vertical, concerne la complexité des actions faites par l'utilisateur.

Une complexité d'interaction faible correspond à de petites tâches qu'il est généralement facile d'accomplir avec le système, tel que commander des fleurs sur un site web. Une haute complexité d'interaction est souvent associée à des tâches plus grandes et plus difficiles, qui requièrent un profil spécialisé ou formé, tel que la manipulation des photos avec Photoshop.

Sur l'axe horizontal de la Figure 6.3, nous pouvons voir la complexité du domaine de travail, qui est le degré de complexité et la nature technique du domaine de travail correspondant. [UX BOOK 2012]

Il faudra donc en tenir compte lors du choix de l'échantillon d'utilisateur.

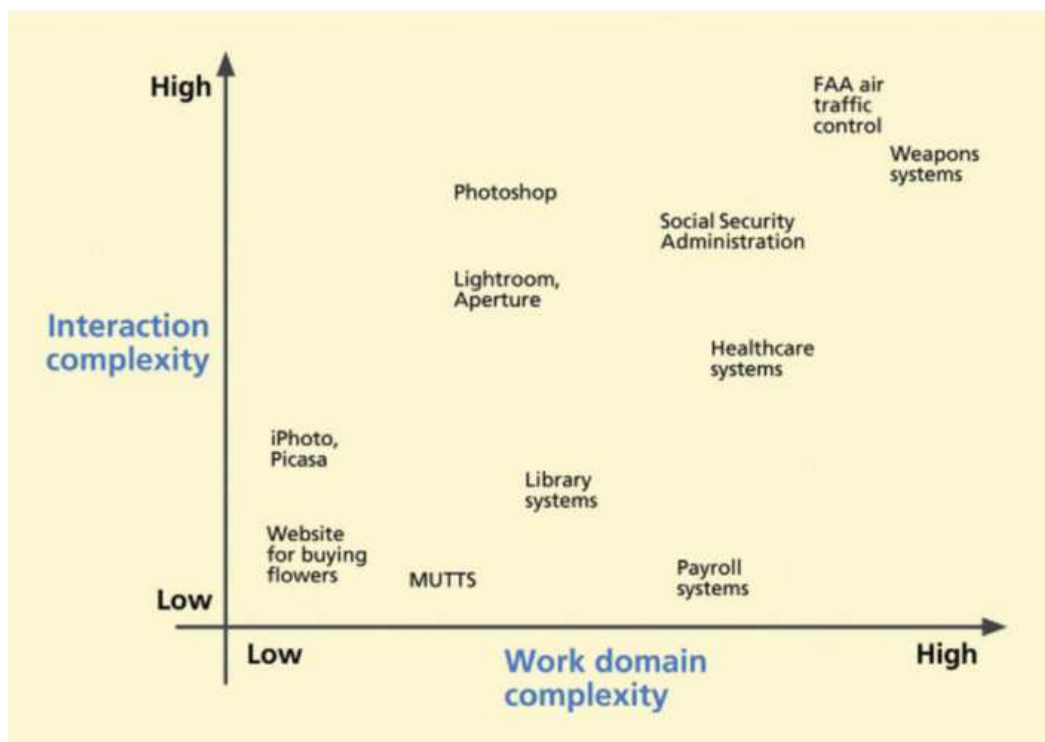


Figure 6.3 : Example systems within the system complexity space (interaction complexity vs. domain complexity)

ISO 9126-2 définit des relations entre les types de métriques. Ainsi, comme le montre la Figure 6.4, des mesures peuvent être prises à différentes étapes du cycle de vie et les attributs de qualité d'une étape vont avoir une influence sur ceux de l'étape suivante.

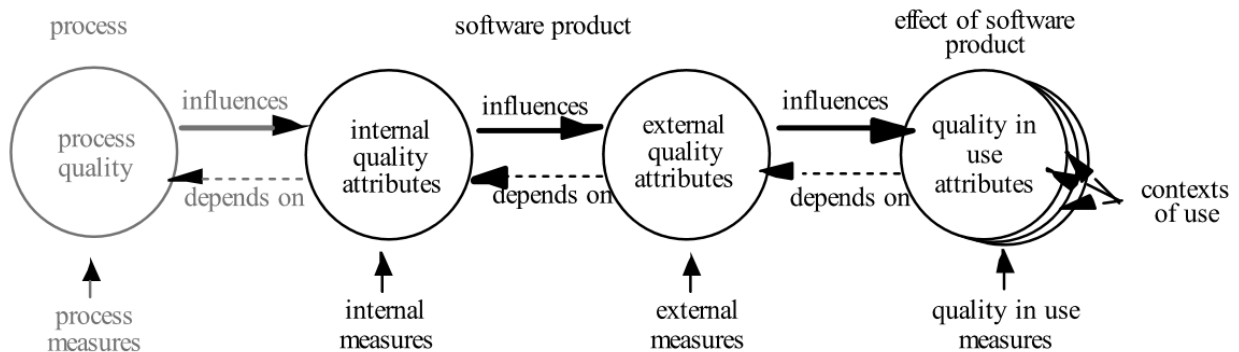


Figure 6.4 : ISO 9126-1 Relationship between types of metrics

Les métriques internes peuvent être relevées sur un produit non-exécutable (code, documentations, analyses, ...) pendant les étapes de développement¹. Les mesures internes permettent de mesurer la qualité des livrables intermédiaires et ainsi prédire la qualité du produit final. Elles permettent d'identifier les problèmes et de planifier les actions correctives aussi tôt que possible dans le cycle de développement.

Les métriques externes peuvent être utilisées pour mesurer la qualité du produit logiciel en mesurant le comportement du système dont il fait partie. Ces métriques externes ne peuvent être mesurées que pendant les étapes de tests du cycle de vie et pendant toutes phases opérationnelles. La mesure est effectuée pendant l'exécution du produit logiciel dans l'environnement système pour lequel il est destiné.

Les mesures de qualité à l'usage mesurent si le produit permet aux utilisateurs spécifiés de répondre à leurs besoins avec efficacité, efficacité et satisfaction dans un contexte précis. Ces mesures ne peuvent être réalisées que durant une utilisation réelle du système.

¹ Par exemple, l'outil SonarQube permet de faire de l'analyse statique de code et donne donc des mesures internes du produit. SonarQube se base sur ISO 9126 et ISO 25010. - <http://www.sonarqube.org/>

6.1.1 Les métriques de compréhensibilité

Les utilisateurs doivent être en mesure de sélectionner un produit logiciel, qui est adapté à l'usage prévu. Une métrique de compréhensibilité devrait être en mesure d'évaluer si les nouveaux utilisateurs peuvent comprendre :

- Si le logiciel est adapté ;
- Comment il peut être utilisé pour des tâches particulières.

Exemple de métrique externe :

Metric name	<i>Completeness of description</i>		
Purpose of the metrics	What proportion of function (or types of functions) is understood after reading the product description?		
Method of application	Conduct user test and interview user with questionnaires or observe user behaviour. Count the number of functions which are adequately understood and compare with the total number of functions in the product.		
Measurement, formula and data element computations	X = A/B A = Number of functions (or types of functions) understood B = Total number of functions (or types of functions)	Target audience	User, Maintainer
Interpretation of measured value	0 ≤ X ≤ 1 The closer to 1.0 is the better.	Metric scale type	Absolute
Measure type	A = Count B = Count X = Count/Count	Input to measurement	User manual Operation (test) report
Footnote	This indicates whether potential users understand the capability of the product after reading the product description.		

[ISO/IEC 9126-2]

Exemple de métrique interne :

Metric name	<i>Completeness of description</i>		
Purpose of the metrics	What proportion of function (or types of functions) are described in the product description?		
Method of application	Count the number of functions which are adequately described and compare with the total number of functions in the product.		
Measurement, formula and data element computations	X = A/B A = Number of functions (or types of functions) described in the product description B = Total number of functions (or types of functions)	Target audience	Requirers, Developers
Interpretation of measured value	0 ≤ X ≤ 1 The closer to 1.0 is the better.	Metric scale type	Absolute
Measure type	A = Count B = Count X = Count/Count	Input to measurement	Req spec Design Review report
Footnote	This indicates whether potential users understand the capability of the product after reading the product description.		

[ISO/IEC 9126-3]

6.1.2 Les métriques de facilité d'apprentissage

Une métrique de facilité d'apprentissage devrait être en mesure d'évaluer combien de temps prend un utilisateur pour apprendre à utiliser une fonctionnalité particulière ainsi que l'efficacité de l'aide ou documentation fournie par le système.

La facilité d'apprentissage est fortement liée à la compréhensibilité, et les mesures de compréhensibilité peuvent être un indicateur du potentiel d'apprentissage du logiciel.

Exemple de métrique externe:

Metric name	<i>Ease of learning to perform a task in use</i>		
Purpose of the metrics	How long does the user take to learn how to perform the specified task efficiently?		
Method of application	Observe user behaviour from when they start to learn until they begin to operate efficiently.		
Measurement, formula and data element computations	T = Sum of user operate time until user achieved to perform the specified task within a short time	Target audience	User, Maintainer
Interpretation of measured value	0<T The short is the better.	Metric scale type	Ratio
Measure type	T = Time	Input to measurement	User manual Operation (test) report User monitoring record
Footnote	It is recommended to determine an expected user's operating time as a short time. Such user's operating time may be the threshold, for example, which is 70% of time at the first use as the fair proportion. Effort may alternatively represent time by person-hour unit.		

[ISO/IEC 9126-2]

Exemple de métrique interne:

Metric name	<i>Completeness of user documentation and/or help facility</i>		
Purpose of the metrics	What proportion of functions are described in the user documentation and/or help facility?		
Method of application	Count the number of functions implemented with help facility and/or documentation and compare with the total number of functions in product.		
Measurement, formula and data element computations	X = A/B A = Number of functions described B = Total of number of functions provided	Target audience	Requirers Developers
Interpretation of measured value	0<=X<=1 The closer to 1.0 is the better.	Metric scale type	Absolute
Measure type	X = Count/Count A = Count B = Count	Input to measurement	Req spec Design Review report
Footnote	Three metrics are possible: completeness of the documentation, completeness off the help facility or completeness of the help and documentation used in combination.		

[ISO/IEC 9126-3]

6.1.3 Les métriques d'opérabilité

Une métrique d'opérabilité devrait être en mesure d'évaluer si les utilisateurs peuvent exploiter et contrôler le logiciel.

Les métriques d'opérabilité peuvent être catégorisées par les principes de dialogue défini dans ISO 9241-10 :

- Adaptation à la tâche
- Aptitude à l'individualisation
- Caractère autodescriptif
- Contrôle utilisateur
- Conformité aux attentes de l'utilisateur
- Tolérance à l'erreur

Le choix des fonctionnalités à tester peut être influencé par la fréquence d'utilisation, la criticité, ou encore d'autres problèmes anticipés d'utilisabilité.

Exemple de métrique externe :

Metric name	<i>Error correction</i>		
Purpose of the metrics	Can user easily correct error on task?		
Method of application	Conduct user test and observe user behaviour.		
Measurement, formula and data element computations	$T = T_c - T_s$ T_c = Time of completing correction of specified type errors of performed task T_s = Time of starting correction of specified type errors of performed task	Target audience	User, Human interface designer
Interpretation of measured value	$0 < T$ The short is the better.	Metric scale type	Ratio
Measure type	T_s, T_c = Time T = Time	Input to measurement	User manual Operation (test) report User monitoring record
Footnote	User of this metric is suggested to specify types of errors for test cases by considering, for example, severity (displaying error or destroying data), type of input/output error (input text error, output data error to database or graphical error on display) or type of error operational situation (interactive use or emergent operation).		

[ISO/IEC 9126-2]

Exemple de métrique interne:

Metric name	<i>Input validity cheching</i>		
Purpose of the metrics	What proportion of input items provide check for valid data?		
Method of application	Count the number of input items, which check for valid data and compare with the number of input items, which could check for valid data.		
Measurement, formula and data element computations	X = A/B A = Number of input items which check for valid data B = Number of input which could check for valid data	Target audience	Requirers Developers
Interpretation of measured value	0<=X<=1 The closer to 1.0 is the better.	Metric scale type	Absolute
Measure type	X = Count/Count A = Count B = Count	Input to measurement	Req spec Design Review report
Footnote	Either single undoability or multiple undoability after several subsequent actions can be assessed.		

[ISO/IEC 9126-3]

6.1.4 Les métriques d'attraction

Une métrique qui mesure si le logiciel est attrayant va évaluer l'apparence de celui-ci et il sera influencé par différents facteurs tels que le design des écrans et les couleurs choisies.

Cet aspect est particulièrement important pour les logiciels « grand public ».

Exemple de métrique externe :

Metric name	<i>Attractive interaction</i>		
Purpose of the metrics	How attractive is the interface to the user?		
Method of application	Questionnaire to users.		
Measurement, formula and data element computations	Questionnaire to assess the attractiveness of the interface to users, after experience of usage.	Target audience	User, Human interface designer
Interpretation of measured value	Depend on its questionnaire scoring method.	Metric scale type	Absolute
Measure type	Count	Input to measurement	Questionnaire result
Footnote	This metric is generally used as one of experienced and justified.		

[ISO/IEC 9126-2]

Exemple de métrique interne :

Metric name	<i>Attractive interaction</i>		
Purpose of the metrics	How attractive is the interface to the user?		
Method of application	Questionnaire to users.		
Measurement, formula and data element computations	Questionnaire to assess the attractiveness of the interface to user, taking account of attributes such as colour and graphical design.	Target audience	Requirers Developers
Interpretation of measured value	Assessment classification.	Metric scale type	Ordinal
Measure type	X = Count (Count is a score)	Input to measurement	Req spec Design Review report
Footnote	This could be based on screen sketches or mock-ups.		

[ISO/IEC 9126-3]

6.1.5 Les métriques de respect de l'utilisabilité

Cette métrique devrait être capable d'évaluer le respect de standards, conventions, guide ou réglementation en relation avec l'utilisabilité.

Exemple de métrique externe :

Metric name	<i>Usability compliance</i>		
Purpose of the metrics	How completely does the software adhere to the standards, conventions, style guides or regulations relating to usability?		
Method of application	Specify required compliance items based on standards, conventions, style guides or regulations relating usability. Design test cases in accordance with compliance items. Conduct functional testing for these test cases.		
Measurement, formula and data element computations	X = $1 - A/B$ A = Number of usability compliance items specified that have not been implemented during testing B = Total number of usability compliance items specified	Target audience	Supplier, User
Interpretation of measured value	$0 \leq X \leq 1$ The closer to 1.0 is the better.	Metric scale type	Absolute
Measure type	A = Count B = Count X = Count/Count	Input to measurement	Product description (User manual or Specification) of compliance and related standards, conventions, style guides or regulations Test specification and report
Footnote	It may be useful to collect several measured values along time, to analyse the trend of increasingly satisfied compliance items and to determine whether they are fully satisfied or not.		

[ISO/IEC 9126-2]

Exemple de métrique interne :

Metric name	<i>Usability compliance</i>		
Purpose of the metrics	How compliant is the product to applicable regulations, standards and conventions for usability?		
Method of application	Count the number of items requiring compliance that have been met and compare with the number of items requiring compliance as in the specification.		
Measurement, formula and data element computations	X = A/B A = Number of correctly implemented items related to usability compliance confirmed in evaluation B = Total number of compliance items	Target audience	Requirers Developers
Interpretation of measured value	Assessment classification.	Metric scale type	Absolute
Measure type	$0 \leq X \leq 1$ The closer to 1.0 is the better.	Input to measurement	Specification of compliance and related standards, conventions or regulations. Design Source code Review report

Footnote

[ISO/IEC 9126-3]

6.2 Critique des métriques d'ISO 9126

Nous venons de voir quelques exemples de ce que ISO 9126-2 et ISO 9126-3 proposent comme mesures pour évaluer l'utilisabilité d'un produit.

Le modèle de qualité d'ISO 9126 sépare la qualité interne et la qualité externe d'un logiciel. Une de ces critiques, c'est que les divisions proposent des méthodes de mesures d'une part de la qualité des artefacts logiciels et du logiciel en test et de l'autre la qualité du logiciel s'exécutant dans l'environnement. Ces méthodes sont trop liées à l'organisation des projets et aux contingences des entreprises.

Le lecteur est aussi en droit de se demander si ces mesures sont réellement possibles dans un contexte réel ? Ce qui est décrit comme « L'opposition entre praticiens et théoriciens, qui est souvent avancée comme un élément qui freine le progrès du génie logiciel », par Ivan Maffezzini, Alice Premiana et Bernardo Ventimiglia dans leur article, ne devrait pas être le cas ici puisque les normes ISO sont conçues par ceux qui les utilisent. Pourtant la mise en place, l'évaluation et l'interprétation des mesures demandent un effort considérable. [QUALITY&METRIC]

La principale difficulté des métriques concernant l'utilisabilité est qu'il est très difficile d'automatiser les méthodes de mesure, ce qui va engendrer un coût gigantesque si nous voulons en appliquer un certain nombre. De plus, dans ces mesures, il y a beaucoup de variables indépendantes, induites par le fait qu'il s'agit de données manuelles, sujettes à interprétation. Il est pratiquement impossible de connaître la marge d'erreur et donc de pouvoir répéter la mesure et faire des comparaisons. Et même si c'était possible, il faudrait voir quelle est sa signification par rapport aux objectifs de qualité.

Une autre critique que nous pouvons faire sur les mesures proposées par ISO 9126 est que l'utilisabilité est vue comme un facteur indépendant de la qualité du logiciel et qu'elle est basée sur des attributs du système. Voir chaque caractéristiques indépendamment peut poser des problèmes car il y a toujours des liens existant entre-elles¹.

Cependant, les attributs requis pour évaluer l'utilisabilité d'un système dépendent du type d'utilisateur (grand public, experts, utilisateurs formés, ...), de la nature de la tâche que l'on souhaite réaliser (la complexité du domaine) et de l'environnement. Dans une approche orientée produit, l'utilisabilité est considérée comme un facteur relativement important de la qualité logiciel. Pourtant, ISO 9126 ne décrit pas dans quelle phase du processus de développement logiciel les mesures doivent être prises. Or, il est important d'être capable de relier des mesures au suivi de projet et de cibler les actions correctives à entreprendre.

De plus, ISO 9126 ne donne aucune indication quant à l'utilisation des mesures et des attributs dans l'identification et la classification des risques liés au projet.

¹ L'exemple le plus parlant est celui de la « facilité d'utilisation » et la « sécurité » qui peuvent être diamétralement opposés. Une authentification via un eToken augmente la caractéristique sécurité mais diminue la facilité d'utilisation.

6.3 Tester l'utilisabilité

D'autres méthodes que celles citées dans ISO 9126 permettent d'avoir un feedback des utilisateurs sur l'utilisabilité d'un système ou logiciel.

Les tests d'utilisabilités impliquent généralement une série de tâches qu'un groupe d'utilisateurs ciblés (idéalement entre 3 et 5) doit accomplir en notant les problèmes rencontrés. Le choix des utilisateurs est très important car il doit constituer un ensemble représentatif des utilisateurs finaux.

Ces méthodes permettent principalement de remonter les problèmes d'utilisabilité mais elles serviront également pour l'évaluation de l'utilisabilité d'un système.

Nous allons citer quelques méthodes permettant de relever des problèmes d'ergonomie ou d'utilisabilité.

6.3.6 Questionnaires

Des questionnaires ont longtemps été utilisés pour évaluer les interfaces utilisateur. Les résultats de cette méthode peuvent être vagues et sujets à interprétation.

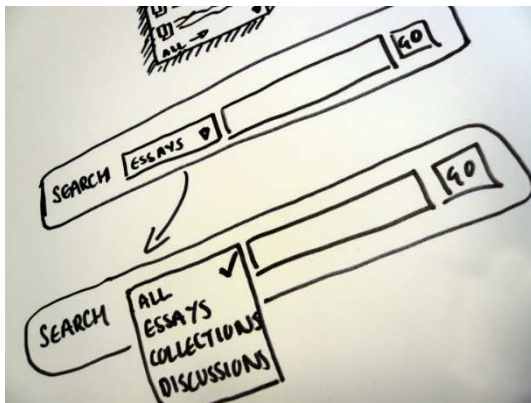
Pour réduire ces risques, nous pouvons utiliser des questionnaires standardisés qui ont été spécifiquement conçus pour évaluer la convivialité et la satisfaction des produits et systèmes informatisés (SUMI, QUIS, etc.).

6.3.7 Think aloud protocol

Comme son nom l'indique, il faut définir une série de tâches et demander aux participants d'énoncer, à haute voix, leurs pensées et sentiments au fur et à mesure qu'ils entrent en contact avec le système.

Le but de cette méthode est d'obtenir des informations sur le processus de pensée au travers des actions réalisées par les participants.

6.3.8 Prototypage



Nous pouvons effectuer des tests de tout ce que nous voulons en utilisant des prototypes papiers. Cette technique est très utilisée dans la conception centrée sur l'utilisateur et aide les parties prenantes d'un projet (concepteurs, développeurs, utilisateurs, ...) à concevoir un logiciel ou un site web qui répond aux attentes et besoins des utilisateurs finaux.

Il s'agit le plus souvent des premières idées brutes jetées sur papier, faites d'esquisses et de dessins, représentant l'agencement et les zones de contenus.

En fonction du besoin (et du budget disponible), différents prototypes avec différents niveaux de détails peuvent être réalisés : esquisse papier, wireframes, visual comps, ...

6.3.9 Eye Tracking

L'Eye Tracking (ou test d'oculométrie) permet de connaître de manière précise et objective la perception réelle qu'ont les utilisateurs d'un système. L'outil mesure la position et les mouvements du regard grâce à un champ de lumière infra-rouge qui éclaire les yeux et dont le reflet est capturé par des caméras.

Ces tests sont généralement réalisés dans une salle de test équipée d'un Eye Tracker permettant d'enregistrer le regard de manière précise et sans aucune gêne pour l'utilisateur.

Cette technique donne une mesure objective de la visibilité des éléments d'une interface, la compréhension des dits éléments, leur impact affectif sur l'utilisateur et enfin, leur impact sur l'exploration de contenus.

Les données enregistrées (temps de fixation du regard sur une zone donnée, ordre du parcours visuel sur une page, ...) peuvent être rejouées et donnent lieu à des analyses statistiques.

Conclusion

En génie logiciel, l'importance de la qualité est reconnue et le problème de produire des logiciels de qualité est étudié par beaucoup de chercheurs. L'importance de l'ergonomie et de l'utilisabilité des logiciels prend également de plus en plus de place dans le cycle de vie logiciel.

Pourtant, encore aujourd'hui, il est difficile de trouver une définition unique de la notion d'utilisabilité. Les rédacteurs des normes ISO s'appuient en partie sur la définition donnée par Shakel (1991), qui définit l'utilisabilité comme « sa capacité, en termes fonctionnels humains, à permettre une utilisation facile et effective par une catégorie donnée d'utilisateurs, avec une formation et un support adapté, pour accomplir une catégorie donnée de tâches, à l'intérieur d'une catégorie spécifique de contextes ». Nous retiendrons surtout les trois composantes initiales données dans la définition d'ISO (efficacité, efficience, satisfaction), même si le contour de cette notion a évolué, y compris à l'intérieur des normes ISO.

Le présent mémoire porte sur une revue critique des normes de qualité logicielle en se concentrant sur l'utilisabilité. La démarche entreprise pour cette recherche vise à rencontrer trois objectifs.

Le premier objectif de ce mémoire est de faire le point sur les documents normatifs de l'Organisation Internationale de Normalisation autour de l'utilisabilité et de la qualité logiciel. Ceux-ci étant nombreux et plus ou moins récents, nous avons pu mieux comprendre l'utilité et la place qu'ils prennent dans le cycle de vie logiciel. Nous avons pu retracer l'historique de ces normes, voir quelles normes sont devenues obsolètes et lesquelles sont remplacées.

Le deuxième objectif est d'explorer les liens entre ISO 9241 et SQuaRE afin de démontrer la pertinence ou non des relations entre les modèles de qualité proposés dans ISO 25010 et les caractéristiques qu'ISO 9241 propose. Le but recherché est de montrer dans quelle mesure SQuaRE a tenu compte d'ISO 9241 pour définir les caractéristiques relatives à l'utilisabilité. A ma connaissance, ce genre d'approche n'a encore jamais été proposé. Ce mémoire apporte donc une première contribution dans ce domaine en proposant un croisement entre SQuaRE et ISO 9241.

Le modèle que nous avons identifié est proposé dans le point « 5.3 - Mapping entre ISO 9241 et ISO 25000 » et montre les liens entre les deux normes citées. Le choix des caractéristiques d'ISO 25010 liés à l'utilisabilité est pris en croisant les deux normes ou en faisant des hypothèses. Il est donc parfaitement discutable que d'autres caractéristiques, non prise en compte, du modèle présenté dans ISO 25010 puissent avoir une influence sur l'utilisabilité.

Enfin, le dernier objectif est d'identifier les mesures proposées pour permettre d'établir des indicateurs basés sur les exigences du guide SQuaRE.

Malheureusement, les mesures décrites dans le présent mémoire ne sont pas celles que SQuaRE propose car les parties ISO 25023 et ISO 25024 ne sont pas encore publiées. Néanmoins, lorsque ces normes sortiront, il serait vraiment très intéressant de les comparer avec les parties 2, 3 et 4 d'ISO 9126.

En effet, les mesures proposées par ISO 9126 ont déjà été beaucoup critiquées car elles sont souvent perçues comme étant trop abstraites ou difficiles à mettre en place (ISO 9126 décrit plus le QUOI que le COMMENT).

Le sujet choisi étant très large, il est difficile de donner l'ensemble des pistes de recherche futures.

Cependant, nous allons en citer quelques-unes :

- Comme déjà discuté ci-dessus, nous pourrions parcourir les méthodes de mesure proposées par SQuaRE lorsqu'elles seront disponibles ;
- Nous pourrions également sortir de SQuaRE et comparer les autres méthodes de mesure de l'utilisabilité d'un logiciel avec celles proposées ;

- Nous pourrions nous poser la question suivante : « Est-ce que les métriques orientent l'interprétation dans certains cas ? » ;
- Nous pourrions mettre en pratique les mesures proposées dans un environnement réel afin de valider que l'approche proposée permet d'augmenter la qualité d'un système du point de vue de l'utilisabilité.

Références

Bibliographie

- [ISO/IEC 25000] ISO/IEC 25000 : *Software engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Guide to SQuaRE*, 2014
- [ISO/IEC 9241-11] ISO/IEC 9241-11 : *Exigences ergonomiques pour travail de bureau avec terminaux à écrans de visualisation (TEV) -- Partie 11: Lignes directrices relatives à l'utilisabilité*, 1998
- [ISO/IEC 9241-110] ISO/IEC 9241-110 : *Ergonomie de l'interaction homme-système -- Partie 110: Principes de dialogue*, 2006
- [ISO/IEC 9241-210] ISO/IEC 9241-210 : *Ergonomie de l'interaction homme-système -- Partie 210: Conception centrée sur l'opérateur humain pour les systèmes interactifs*, 2010
- [ISO/IEC 9126-2] ISO/IEC TR 9126-2 : *Software engineering – Product quality – Part 2 : External metrics*, 2003
- [ISO/IEC 9126-3] ISO/IEC TR 9126-3 : *Software engineering – Product quality – Part 3 : Internal metrics*, 2003
- [ISO/IEC 9126-4] ISO/IEC TR 9126-3 : *Software engineering – Product quality – Part 4 : Quality in use metrics*, 2004
- [ISO 1997] Lawrence D. Eicher, Willy Kuert, Roger Maréchal, Vince Grey, Raymond Frontard, Olle Sturen, Anders Thor, Roseline Barchietto : *L'histoire d'une amitié partagée – Souvenirs à propos des cinquante premières années de l'ISO*, 2012
- [Ortega, Pérez & Rojas 2003] Maryoly Ortega, María Pérez, Teresita Rojas: *Construction of a Systemic Quality Model for Evaluating a Software Product*, 2003
- [UX BOOK 2012] Rex Hartson, Pardha s. Pyla: *The UX Book - Process and Guidelines for Ensuring a Quality User Experience*, 2012
- [HCI 2006] Nigel Bevan, *International Standards for HCI*, 2006
- [IHM 2015-2016] Bruno Dumas : *Human Machine Interaction (HD)*, 2015-2016
- [SQJ 2003] A Abran, A Khelifi, W Suryn, A Seffah: *Software Quality Journal*, 2003 - Springer
- [CPFU 2003] Éric BRANGIER et Javier BARCENILLAA: *Concevoir un produit facile à utiliser – Adapter les technologies à l'homme*, 2003
- [FOCUS+ 2010] ISO: *ISO Focus+*, Le magazine de l'Organisation Internationale de normalisations, Vol. 1, No. 1, janvier 2010, ISSN 1729-8709, p.7

- [FOCUS+ 2007] ISO : ISO Focus+, Le magazine de l'Organisation Internationale de normalisations, Vol. 4, No. 5, May 2007, p.38-42
- [QUALITY&USABILITY] Nigel Bevan: Quality and usability: A new framework, 1997
- [QUALITY&METRIC] Ivan Maffezzini, Alice Premiana et Bernardo Ventimiglia : Prolégomènes à une critique du génie logiciel - Partie II : Qualité et mesures des produits

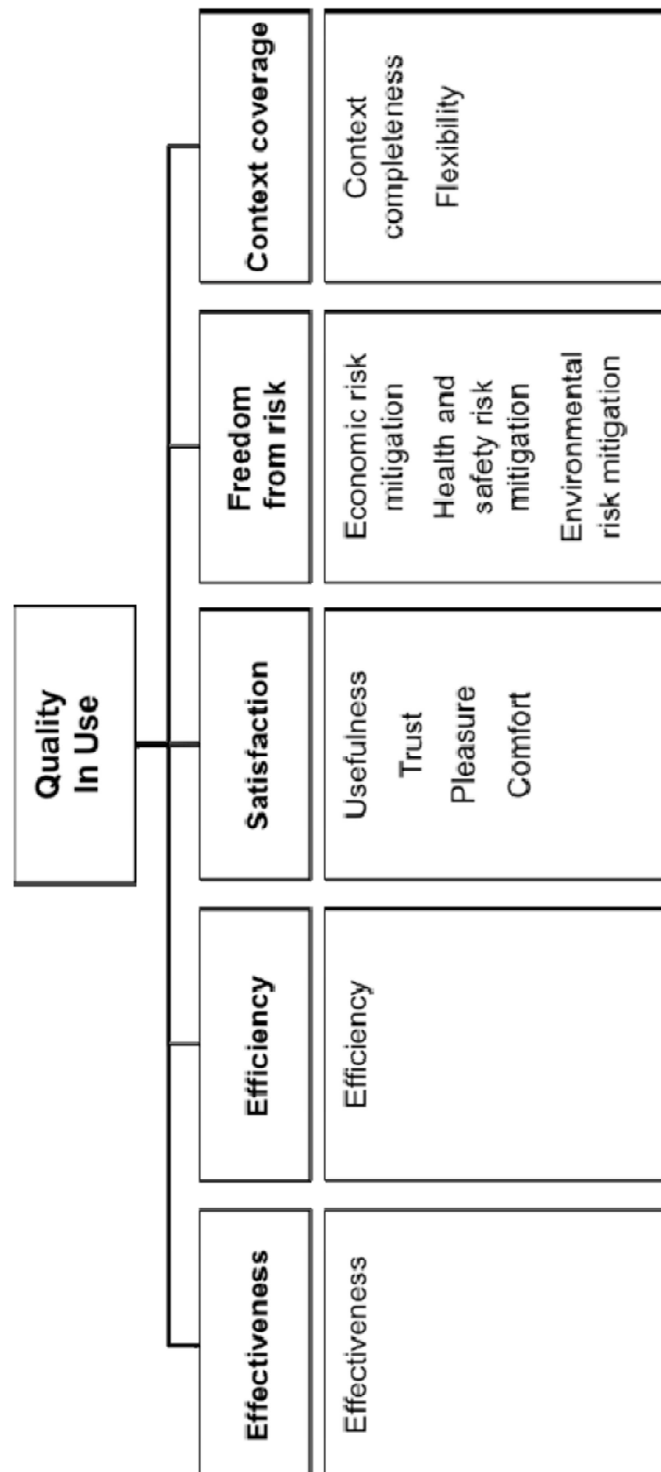
Webographie

- Organisation Internationale de Normalisation : <http://www.iso.org/iso/fr/>
- Organisation Internationale de Normalisation : *Il était une fois l'iso*, http://www.iso.org/iso/fr/home/about/the_iso_story.htm (consulté le 06/11/2015).
- Jakob Nielsen, 10 Usability Heuristics for User Interface Design <http://www.nngroup.com/articles/ten-usability-heuristics/> (consulté le 25/11/2015).

Annexe 1 : SQuaRE - Quality In Use

The quality in use model defines five characteristics related to outcomes of interaction with a system : effectiveness, efficiency, satisfaction, freedom from risk, and context coverage. Each characteristic can be assigned to different activities of stakeholders, for example, the interaction of an operator or the maintenance of a developer.

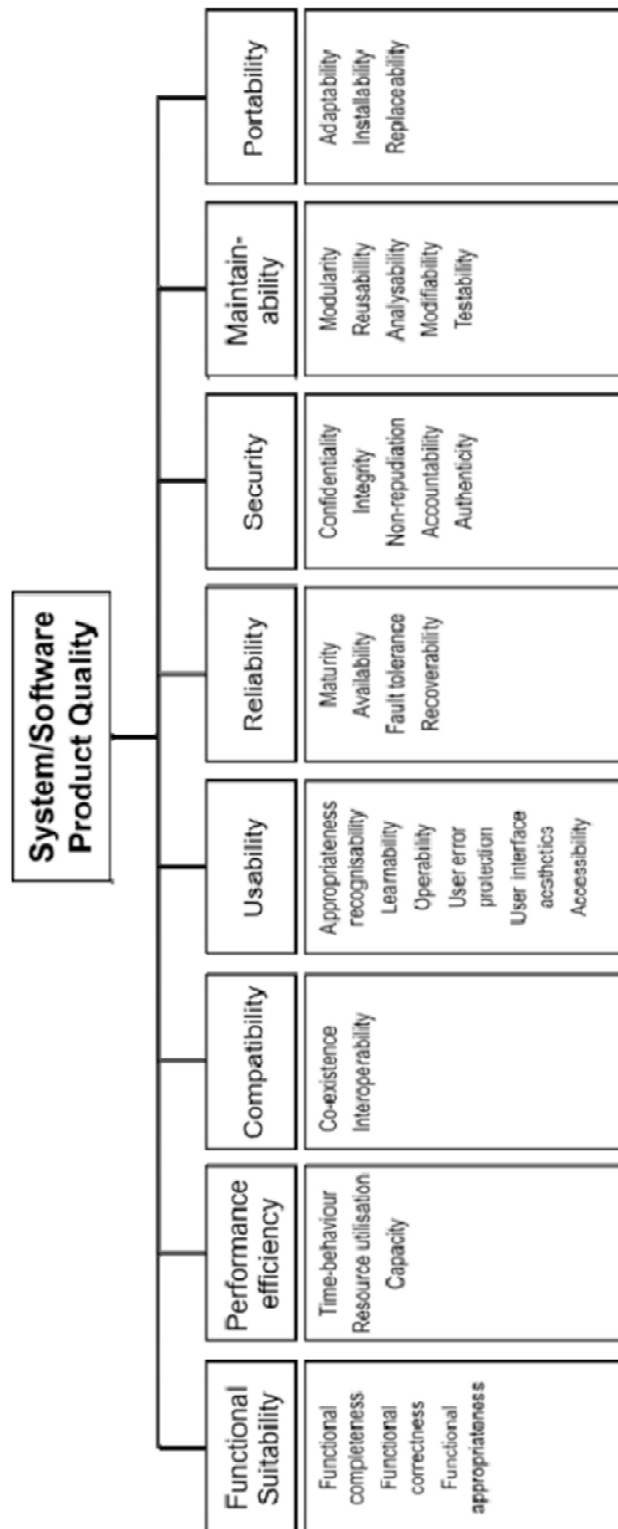
The quality in use of a system characterizes the impact that the product (system or software product) has on stakeholders. It is determined by the quality of the software, hardware and operating environment, and the characteristics of the users, tasks and social environment. All these factors contribute to the quality in use of the system.



Annexe 2 : SQuaRE - System/Software Product Quality

The product quality model categorizes system/software product quality properties into eight characteristics: functional suitability, performance efficiency, compatibility, usability, reliability, security, maintainability and portability. Each characteristic is composed of a set of related subcharacteristics.

The product quality model can be applied to just a software product, or to a computer system that includes software, as most of the subcharacteristics are relevant to both software and systems.



Annexe 3 : Rapport sur la qualité d'un produit

Qualité à l'usage		2,00
Efficacité		2,00
Efficience		1,00
Satisfaction		2,00
Utilité		3
Confiance		2
Plaisir		1
Confort		2
Absence de risque		
Atténuation des risques économiques		
Atténuation des risques liés à l'intégrité des personnes		
Atténuation des risques environnementaux		
Couverture du contexte		2,00
Complétude du contexte		3
Flexibilité		1

Qualité produit		2,67
Adéquation fonctionnelle		2,67
Complétude		3
Exactitude		3
Pertinence		2
Performance		
Comportement dans la durée		
Utilisation des ressources		
Capacité		
Compatibilité		
Coexistence		
Interopérabilité		
Facilité d'utilisation		1,33
Facilité d'apprentissage		2
Opérabilité		1
Protection de l'utilisateur à faire des erreurs		2
Esthétique		0
Accessibilité		1

Qualité produit		2,00
Fiabilité		
Maturité		
Disponibilité		
Tolérance aux pannes		
Capacité de récupération		
Sécurité		
Confidentialité		
Intégrité		
Non-répudiation		
Responsabilité		
Authenticité		
Maintenabilité		
Modularité		
Réutilisabilité		
Analysabilité		
Modificabilité		
Testabilité		
Portabilité		2,00
Adaptabilité		2
Capacité de déploiement		
Capacité de remplacement		

Légende	
Pas pris en compte	
0	-
1	--
2	-
3	+

